

CT_EX 宏集手册

CTEX.ORG

2020/05/02 v2.5.1*

简介

CT_EX 宏集是面向中文排版的通用 L^AT_EX 排版框架,为中文 L^AT_EX 文档提供了汉字输出支持、标点压缩、字体字号命令、标题文字汉化、中文版式调整、数字日期转换等支持功能,可适应论文、报告、书籍、幻灯片等不同类型的中文文档。

CT_EX 宏集支持 L^AT_EX、pdfL^AT_EX、X_YL^AT_EX、LuaL^AT_EX、upL^AT_EX 等多种不同的编译方式,并为它们提供了统一的界面。主要功能由宏包 `ctex` 以及中文文档类 `ctexart`、`ctexrep`、`ctexbook` 和 `ctexbeamer` 实现。

目录

第 1 节 介绍	2	第 7 节 章节标题样式设置	15
第 2 节 简明教程	3	7.1 编号相关	16
2.1 CT _E X 宏集的组成	3	7.2 格式相关	17
2.2 CT _E X 宏集的安装和更新	3	7.3 间距、缩进相关	21
2.3 使用 CT _E X 文档类	4	7.4 目录、附录相关	24
2.4 使用 <code>ctex</code> 宏包	5	7.5 有关命令	25
第 3 节 宏包选项与 <code>\ctexset</code> 命令	5	7.6 示例	26
第 4 节 编译方式、编码与中文字库	6	第 8 节 实用命令	27
4.1 编译方式	6	8.1 字号与间距	27
4.2 中文编码	6	8.2 中文数字转换	27
4.3 中文字库	6	8.3 杂项	28
第 5 节 排版格式设定	8	第 9 节 LuaL ^A T _E X 下的中文支持方式	28
5.1 文档默认字号	8	9.1 LuaL ^A T _E X 下替代字体的设置	29
5.2 章节标题风格	9	第 10 节 CT _E X 宏集的配置文	30
5.3 排版方案选项	10	10.1 修改宏包默认选项	30
第 6 节 文档汉化	11	10.2 宏包载入后的配置	31
6.1 日期汉化	11	10.3 配置标题中文翻译	31
6.2 文档标题汉化	12	10.4 自定义字体集	31
6.3 页面格式设置与汉化	13	第 11 节 对旧版本的兼容性	31
		11.1 CT _E X 0.8a 及以前的版本	31
		11.2 CT _E X 0.9–CT _E X 1.0d	32
		11.3 CT _E X 1.02c 以后的 SVN 开发版	33
		11.4 CT _E X 2.2 之前的版本	34
		11.5 CT _E X 2.4.1 和 2.4.2	34

*`ctex-kit` rev. 320df30.

11.6 C _T E _X 2.5 之前的版本	34	第 14 节 代码实现	37
第 12 节 宏集依赖情况与手工安装方法	35	版本历史	143
第 13 节 开发人员	36	代码索引	145

第 1 节 介绍

历史

C_TE_X 宏集的源头有两个：一是王磊编写的 `cjkbook` 文档类，二是吴凌云编写的 `GB.cap`。这些工作没有经过认真系统的设计，也没有用户文档，不利于维护和改进。

2003 年，吴凌云使用 `doc` 和 `DocStrip` 重构了整个工程，并增加了许多新的功能，称为 `ctex` 宏包。2007 年，oseen 和王越在 `ctex` 宏包的基础上，增加了对 UTF-8 编码的支持，开发出了 `ctexutf8` 宏包。

2009 年 5 月，我们在 Google Code 建立了 `ctex-kit` 项目¹，对 `ctex` 宏包及相关脚本进行了整合，并加入了对 X_YL_AT_EX 引擎的支持。在开发新版本时，考虑到合作开发和调试的方便，我们放弃了 `doc` 和 `DocStrip`，采取了直接编写宏包代码的方式。

2014 年 3 月，为了适应 L_AT_EX 的最新发展，特别是 L_AT_EX3 的逐渐成熟，李清用 L_AT_EX3 重构了整个宏包的代码，并重新使用 `doc` 和 `DocStrip` 工具进行代码的管理，升级版本号为 2.0，并改称 C_TE_X 宏集。

2015 年 3 月，由于 Google Code 即将停止服务，`ctex-kit` 项目迁移至 [GitHub](#)²。

最初，Knuth 在设计开发 T_EX 的时候没有考虑到多国文字支持，特别是对多字节的中日韩表意文字的支持。这使得 T_EX 以至后来的 L_AT_EX 对中文的支持一直不是很好。即使在 CJK 宏包解决了中文字符处理的问题以后，中文用户使用 L_AT_EX 仍然要面对许多困难。这些困难里，以章节标题的中文化为最。由于中文和西文书写习惯的差异，用户很难使用标准文档类中的代码结构来表达中文标题。于是，用户不得不对标准文档类做较大的修改。除此之外，日期格式、首行缩进、中文字号和字距等细节问题，也需要精细的调校。我们设计 C_TE_X 宏集的目的之一就是解决这些 L_AT_EX 文档的汉化难题。

另一方面，随着 T_EX 引擎和 L_AT_EX 宏包的不断发展，L_AT_EX 的中文支持方式从早期的专用系统（如 CCT）发展为适用于不同引擎的多种方式³。这些方式的适用情况和使用方式有不少细节上的差异，同时操作系统的不同、语言环境的不同等客观情况又进一步带来了更多的细节差异。我们设计 C_TE_X 宏集的另一个主要目的就是尽可能消除这些差异带来的影响，使用户能够以一个统一的接口来使用不同的中文支持方式，使得同一份文档能够在不同环境下交换使用。

C_TE_X 宏集的许多实现细节离不开热心朋友们在 [bbs.ctex.org](#) 论坛⁴上的讨论，在此对参与讨论的朋友们表示感谢。

关于宏集名字的说明

C_TE_X 之名是英文单词 China（中国）或 Chinese（中文）的首字母“C”与“T_EX”结合而成的。在纯文本环境下，该名字应写作“CTeX”。

¹<http://code.google.com/p/ctex-kit/>，该链接现已失效。

²<https://github.com/CTeX-org/ctex-kit>

³比如：pdf_TE_X 引擎下的 CJK、zhmCJK 宏包，X_YL_AT_EX 引擎下的 xeCJK 宏包和 Lua_TE_X 引擎下的 Lua_TE_X-ja 宏包。

⁴2018 年，C_TE_X 论坛因故无限期关闭，此链接现已失效。

CT_EX 宏集是由 CT_EX 社区发起并维护的 L^AT_EX 宏包和文档类的集合。社区另有发布名为 CT_EX 套装的 T_EX 发行版,与本文档所述的 CT_EX 宏集并非是一事物。

ctex 则是本宏集中的 ctex.sty 的名字。这一完全小写的名称,在过去也被用来指代整个 CT_EX 宏集,不过现在则特指 ctex.sty 这一宏包。在不引起歧义的情况下,它也可以沿用过去的习惯,代指整个宏集。

第 2 节 简明教程

2.1 CT_EX 宏集的组成

为了适应用户不同的需求,我们将 CT_EX 宏集的主要功能设计安排在四个中文文档类和三个宏包当中,具体的组成见表 1。

表 1 CT_EX 宏集的组成

类别	文件	说明
文档类	ctexart.cls	标准文档类 article 的汉化版本,一般适用于短篇幅的文章
	ctexrep.cls	标准文档类 report 的汉化版本,一般适用于中篇幅的报告
	ctexbook.cls	标准文档类 book 的汉化版本,一般适用于长篇幅的书籍
	ctexbeamer.cls	文档类 beamer 的汉化版本,适用于幻灯片演示
宏包	ctex.sty	提供全部功能,但默认不开启章节标题设置功能,需要使用 heading 选项来开启
	ctexsize.sty	定义和调整中文字号,可以在 ctex 宏包或 CT _E X 中文文档类之外单独调用
	ctexheading.sty	提供章节标题设置功能(见 7 节),可以在 ctex 宏包或 CT _E X 中文文档类之外单独调用

2.2 CT_EX 宏集的安装和更新

最常见的 T_EX 发行版(T_EX Live 和 MiK_T_EX)已收录 CT_EX 宏集及其依赖的宏包和宏集。⁵如果本地安装 T_EX Live 或 MiK_T_EX 不是完整版本,则可能需要通过这两个发行版提供的包管理器来安装宏包。

T_EX Live 的包管理器是 tlmgr(T_EX Live Manager)。用户可以在系统命令行中⁶执行

```
tlmgr gui
```

启动管理器的图形界面(Windows 用户也可以通过开始菜单的 TeX Live 20XX(年份)→TeX Live Manager 打开)。连接上远程仓库之后,搜索 ctex 即可安装。tlmgr 的图形界面使用 Perl 编写,在实践中发现容易造成系统假死。遇到这种问题的用户,也可以直接在系统命令行执行

```
tlmgr install ctex
```

来安装 CT_EX 宏集⁷。

⁵zhmCJK 宏包是个例外。当用户显式指定选项 zhmap = zhmCJK 时,CT_EX 宏集依赖它。由于,它没有被 T_EX Live 和 MiK_T_EX 收录,用户可能需要遵照其说明文档自行安装。

⁶Windows 系统的命令行是 CMD 命令提示符,你可以使用 Win + R 组合键打开“运行”对话框,然后输入 cmd 确认打开命令提示符窗口。

⁷*nix 用户可能需要超级用户权限(sudo)才能正确安装宏集。

MiKTeX 通常会在缺失宏包时自动完成安装。如需手动安装,可以使用其管理维护工具 MiKTeX Console。用户可以打开管理器,连接上远程仓库之后,在“Package”选项卡中搜索“ctex”并安装即可。也可以使用 mpm (MiKTeX Package Manager),在命令行执行

```
mpm --admin --install=ctex
```

来安装 CT_EX 宏集。

若希望了解 CT_EX 宏集具体的依赖情况或手工安装宏集的方法,请参阅第 12 节。

当我们将宏集的新版本发布于 CTAN,且为发行版的远程仓库更新后,用户就可以在本地通过包管理器获取新版本。

对于 T_EX Live,可以在 tlmgr 的图形界面点击“更新全部已安装的”按钮或者在命令行执行

```
tlmgr update --all
```

来完整更新已安装的宏包。

对于 MiKTeX,在 MiKTeX Console 中找到“Updates”选项卡,检查更新后即可选择升级宏包。也可以使用 mpm,在命令行执行

```
mpm --admin --update
```

来进行更新。

2.3 使用 CT_EX 文档类

如果用户需要在三个标准文档类或 *beamer* 的基础上添加中文及版式的支持,我们建议用户使用 CT_EX 宏集提供的四个中文文档类。

CT_EX 宏集提供了四个中文文档类:ctexart、ctexrep、ctexbook 和 ctexbeamer,分别对应 L^AT_EX 的标准文档类 article、report、book 和 beamer。使用它们的时候,需要将涉及到的所有源文件使用 UTF-8 编码保存⁸。

例 1

```
\documentclass{ctexart}
\begin{document}
中文文档类测试。你需要将所有源文件保存为 UTF-8 编码。

你可以使用 XeLaTeX、LuaLaTeX 或 upLaTeX 编译,也可以使用 (pdf)LaTeX 编译。
推荐使用 XeLaTeX 或 LuaLaTeX 编译。对高级用户,我们也推荐使用 upLaTeX 编译。
\end{document}
```

以下是使用 ctexbeamer 文档类编写中文演示文稿的一个示例。

例 2

```
\documentclass{ctexbeamer}
\begin{document}
\begin{frame}{中文演示文档}
\begin{itemize}
\item 你需要将所有源文件保存为 UTF-8 编码
\item 你可以使用 XeLaTeX、LuaLaTeX 或 upLaTeX 编译
\item 也可以使用 (pdf)LaTeX 编译
\end{itemize}
\end{frame}
\end{document}
```

⁸使用 (pdf)L^AT_EX 时也能够使用 GBK 编码,但不推荐。(见 4.2 节)

```

\item 推荐使用 XeLaTeX 或 LuaLaTeX 编译
\item 对高级用户，我们也推荐使用 upLaTeX 编译
\end{itemize}
\end{frame}
\end{document}

```

2.4 使用 ctex 宏包

用户在使用非标准文档类及 *beamer* 时，如果需要添加中文及版式的支持，则可以使用 *ctex* 宏包。

对于建立在 L^AT_EX 标准文档类之上开发的文档类，在使用 *ctex* 宏包时加上 *heading* 选项，可以将章节标题设置为中文风格。

例 3

```

\documentclass{ltxdoc}
\usepackage[heading = true]{ctex}
\begin{document}
\section{简介}
章节标题中文化的 \LaTeX{} 手册。
\end{document}

```

第3节 宏包选项与 \ctexset 命令

CT_EX 宏集已经尽可能就中文的行文和版式习惯做了调整和配置，通常而言，这些配置已经够用。因此，除非必要，我们不建议普通用户修改这些默认配置。如果你认为 CT_EX 宏集的默认配置还可以完善，可以在项目主页上[提交 issue](#)，向我们反映，我们会酌情在后续版本中予以改进。

不过，CT_EX 宏集也提供了一系列选项。用户可以使用这些选项来控制 CT_EX 宏集的行为。按形式分类，这些选项有的以传统的方式提供，有的以 $\langle key \rangle = \langle value \rangle$ 的形式提供。按指定位置分类，这些选项又可以分为以下三类：

- 名字后带有 ☆ 号的选项，只能作为宏包/文档类选项，需要在引入宏包/文档类的时候指定；
- 名字后带有 ★ 号的选项，只能通过 CT_EX 宏集提供的用户接口 \ctexset 来设定；
- 名字后不带有特殊符号的选项，既可以作为宏包/文档类选项，也可以通过 \ctexset 来设定。

后续文档将在使用说明中对某些特殊的选项加以说明。

\ctexset

New: 2014-03-18

\ctexset {<键值列表>}

是 CT_EX 宏集的通用控制命令，用来在宏包载入后控制宏包的各项功能。 \ctexset 的参数是一个键值列表，以通用的接口完成各项设置。

\ctexset 的参数是一组由逗号分隔的选项列表，列表中的选项通常是一个 $\langle key \rangle = \langle value \rangle$ 格式的定义。例如设置摘要与参考文献标题名称(6.2 节)就可以使用：

例 4

```

\ctexset{
  abstractname = {本文概要},
  bibname      = {文\quad 献}
}

```

`\ctexset` 采用 \LaTeX 3 风格的键值设置, 支持不同类型的选项与层次化的选项设置, 相关示例见 7 节。

第 4 节 编译方式、编码与中文字库

4.1 编译方式

\CTeX 宏集会根据用户使用的编译方式⁹, 在底层选择不同的中文支持方式(见表 2)。

表 2 \CTeX 宏集的中文支持方式

编译方式	(pdf) \LaTeX	\XeLaTeX	\LuaLaTeX	\upLaTeX [*]
支持宏包	CJK	xeCJK	LuaTeX-ja	原生

^{*} \pLaTeX-ng (或称 \ApLaTeX) 与 \upLaTeX 兼容。使用 \pLaTeX-ng 编译时, `ctex` 采用与 \upLaTeX 相同的设置。

不同的编译方式和中文支持方式会在一定程度上影响 \CTeX 宏集的行为, 比如宏包对文档编码、字体选择、空格、标点等的处理。具体细节将在本文档后续内容中进行阐述。

4.2 中文编码

GBK	☆	指明编写文档时使用的编码。 \CTeX 宏集无法检测文档源文件的实际编码格式, 因此需要用户通过选项声明。如果没有显式指定, 则默认采用 UTF-8 编码。
UTF8	☆	
Updated: 2019-11-10		

使用 \XeLaTeX 、 \LuaLaTeX 或 \upLaTeX 编译时, \CTeX 宏集强制使用 UTF-8 编码, 此时 GBK 选项无效; 使用 (pdf) \LaTeX 编译时, \CTeX 宏集默认使用 UTF-8 编码, 但用户也可以显式声明 GBK 选项, 使 \CTeX 宏集按 GBK 编码处理文档。

用户需要保证编译方式、源文件编码、宏包编码选项三者一致。

我们建议编写新文档时始终使用 UTF-8 编码, 而仅把 GBK 编码留给历史遗留文档。

4.3 中文字库

以往, 为 \LaTeX 文档配置中文支持是一件相当繁琐的事情。默认情况下, \CTeX 宏集能自动检测用户使用的编译方式(参见 4.1 节)和操作系统¹⁰, 选择合适的底层支持和字库, 从而简化配置过程。自动配置的情况参见表 3。

通常, 由 \CTeX 宏集进行的自动配置已经足够使用, 无需用户手工干预; 但是 \CTeX 仍然提供了一系列选项, 供在 \CTeX 的自动选择机制因为意外情况失效, 或者在用户有特殊需求的情况下使用。除非必要, 用户不应使用这些选项。

zhmap	☆	<code>zhmap = <true false zhmCJK></code>
Updated: 2014-03-08		

指定字体映射机制。本选项只在使用 (pdf) \LaTeX 编译时有意义。

这是该选项的默认值。¹¹使用 `zhmetrics` 宏包, 将 CJK 字库通过 `\special` 命令映射到 .ttf 文件。

⁹ \LaTeX 、pdf \LaTeX 、 \XeLaTeX 、 \LuaLaTeX 及 \upLaTeX 。
¹⁰ \CTeX 宏集现在能够识别 macOS 及 Windows 系统, 并将其他系统统一归为 Linux。
¹¹ 对于以键值对形式提供的选项, 文档以**粗体**来表示其默认值。下同, 不再额外解释。

表 3 CT_EX 宏集自动配置字体策略

	macOS Old ¹	macOS New ²	Windows ³	其他
X _Y L ^A T _E X	xeCJK 华文字库	xeCJK 华文字库 + 苹方	xeCJK 中易字库 + 微软雅黑	xeCJK Fandol 字库 ⁴
LuaL ^A T _E X ⁵	LuaTeX-ja 华文字库	LuaTeX-ja 华文字库 + 苹方	LuaTeX-ja 中易字库 + 微软雅黑	LuaTeX-ja Fandol 字库
pdfL ^A T _E X	不可用	不可用	CJK + zhmetrics 中易字库 + 微软雅黑 ⁶	不可用
L ^A T _E X + DVIPDFM _x	不可用	CJK + zhmetrics 华文字库 + 苹方	CJK + zhmetrics 中易字库 + 微软雅黑 ⁶	CJK + zhmetrics Fandol 字库
upL ^A T _E X + DVIPDFM _x	不可用	zhmetrics-uptex 华文字库 + 苹方	zhmetrics-uptex 中易字库 + 微软雅黑	zhmetrics-uptex Fandol 字库

1 Yosemite (10.10) 及以前的 macOS 系统。
2 El Capitan (10.11) 及以后的 macOS 系统。
3 仅支持 Windows Vista 及以后的 Windows 操作系统。
4 由马起园、苏杰、黄晨成等人开发的开源中文字体, 参见:<https://www.ctan.org/pkg/fandol>。
5 LuaL^AT_EX 编译时使用 LuaTeX-ja 宏包。对此, 第 9 节有特别说明。
6 微软雅黑字体并不总是有效, 这和选项 zhmap 的取值有关。

false 使用传统的 CJK 字库 (Type 1)¹²。
zhmCJK 载入 zhmCJK 宏包¹³, 由 zhmCJK 宏包提供从 CJK 字库到 .ttf 的映射。

fontset	fontset = {adobe fandol founder mac macnew macold ubuntu windows none ...}
New: 2014-03-08	指定 CT _E X 宏集加载的字库。

如果没有指定 fontset 的值, CT_EX 宏集将自动检测用户使用的操作系统, 配置相应的字库 (参见表 3)。

CT_EX 预定义了以下六种中文字库。

adobe	使用 Adobe 公司的四款中文字体, 不支持 pdfL ^A T _E X。
fandol	使用 Fandol 中文字体, 不支持 pdfL ^A T _E X。
founder	使用方正公司的中文字体。
mac	使用 macOS 系统下的字体, 不支持 pdfL ^A T _E X, 根据版本分为 macnew 和 macold 两种。
macnew	使用 El Capitan 或之后的多字重华文字体和苹方字体。
macold	使用 Yosemite 或之前的华文字体。
ubuntu	使用 Ubuntu 系统下的思源宋体、思源黑体和 T _E X 发行版自带的文鼎楷体, 不支持 pdfL ^A T _E X。
windows	使用 Windows 系统下的中易字体和微软雅黑字体。当使用 (pdf)L ^A T _E X 编译时, 微软雅黑仅在以下两种情形有效: 安装有 zhmCJK 宏包且选项 zhmap=zhmCJK 时, 或者安装有微软雅黑的 Type1 字体且选项 zhmap=false 时。

如果不想使用 CT_EX 预定义的中文字库, 可以设置 fontset 为下述值之一。

none	不配置中文字体, 需要用户自己配置。
(name)	这里 (name) 为自定义的名字。CT _E X 宏集将载入名为 ctex-fontset-(name).def 的文件作为字体配置文件。因此, 请先保证文件的存在。可以在当前工作目录或者本地 TDS 目录下

¹²使用 (pdf)L^AT_EX 编译时, 如果需要使用自定义的字体映射文件 (比如需要使用 L^AT_EX + Dvips 编译), 或者希望使用 Type1 字库, 请禁用本选项。为此, 你可能需要安装 CJK 字体。参考 zhmetrics 宏包提供的脚本 [CTeXFonts.lua](#)。
¹³zhmCJK 宏包基于 zhmetrics 和 CJK 宏包, 提供与 xeCJK 宏包类似的用户接口。

合适位置建立一个名为 `ctex-fontset-<name>.def` 的文件, 在这个文件里面自定义中文字体。然后通过使用 `fontset=<name>` 选项来调用它。字体配置文件的具体写法可以参考 C_TE_X 宏集 `fontset` 目录下的字体配置文件。

注意: 如果希望使用 `\ctexset` 在导言区指定字库, 则需要先在宏包/文档类选项中指定 `fontset = none`(这会禁用 C_TE_X 宏集的操作系统检测功能和自动设定字库功能)。例如:

例 5

```
\documentclass[fontset = none]{ctexart}
\ctexset{fontset = founder}
\begin{document}
在文档类选项中声明 \verb|fontset = none|, 随后在导言区用 \verb|\ctexset|
指定字体。
\end{document}
```

C_TE_X 宏集预定义的中文字库还定义了一些字体命令。除了在 `ubuntu` 字库中没有 `\fangsong` 的定义外, 所有字库都有以下四个字体命令:

`\songti` 宋体, CJK 等价命令 `\CJKfamily{zh song}`。
`\heiti` 黑体, CJK 等价命令 `\CJKfamily{zh hei}`。
`\fangsong` 仿宋, CJK 等价命令 `\CJKfamily{zh fs}`。
`\kaishu` 楷书, CJK 等价命令 `\CJKfamily{zh kai}`。

在 `windows`、`founder` 和 `macnew` 字库中, 额外定义了 `\lishu` 和 `\youyuan`:

`\lishu` 隶书, CJK 等价命令 `\CJKfamily{zh li}`。
`\youyuan` 圆体, CJK 等价命令 `\CJKfamily{zh you}`。

在 `windows` 字库中还定义了 `\yahei`。出于兼容性的考虑, `\yahei` 命令在 `macnew` 字库中也有定义, 但实际调用苹方黑体:

`\yahei` 微软雅黑, CJK 等价命令 `\CJKfamily{zh yahei}`。

在 `macnew` 字库中, 还定义了 `\pingfang`:

`\pingfang` 苹方黑体, CJK 等价命令 `\CJKfamily{zh pf}`。

第 5 节 排版格式设定

5.1 文档默认字号

`zihao` ☆ `zihao = <-4|5|false>`

New: 2015-05-06

将文章默认字号 (`\normalsize`) 设置为小四号字或五号字, 具体情况见表 4。 `false` 禁用本功能。本选项可以用于四个 C_TE_X 文档类和 `ctex` 宏包, 也可以用于 `ctexsize` 宏包。

该选项的默认值与 `scheme` 的取值有关。当 `scheme = chinese` 时, 对标准文档类默认值为 5, 即设置 `\normalsize` 为五号字; 对 `beamer` 则为 `false`, 使用文档类原有的设置。当 `scheme = plain` 时, 该选项不设默认值, 沿用标准文档类或 `beamer` 的原有设置。

10pt ☆
11pt ☆
12pt ☆

C_TE_X 文档类是在 L_AT_EX 标准文档类之上开发的。因此, 除了可以使用 C_TE_X 宏包定义的字号选项之外, 还可以使用标准文档类的同类选项 (10pt、11pt 和 12pt)。在使用这些来自标准文档类的选项的时候, C_TE_X 文档类的字号选项会被抑制。亦即, 在 `zihao` 选项之后设置 10pt 选项, `zihao` 选项将不再起作用。

表 4 标准字体命令与字号的对应

字体命令	zihao = 5		zihao = -4		10pt	11pt	12pt
	字号	bp	字号	bp	pt	pt	pt
\tiny	七号	5.5	小六	6.5	5	6	6
\scriptsize	小六	6.5	六号	7.5	7	8	8
\footnotesize	六号	7.5	小五	9	8	9	10
\small	小五	9	五号	10.5	9	10	11
\normalsize	五号	10.5	小四	12	10	11	12
\large	小四	12	小三	15	12	12	14
\Large	小三	15	小二	18	14	14	17
\LARGE	小二	18	二号	22	17	17	20
\huge	二号	22	小一	24	20	20	25
\Huge	一号	26	一号	26	25	25	25

标准文档类的其他选项在 C_TE_X 文档类中依旧有效。例如，设置纸张大小和方向的 a4paper 和 landscape，设置单双面的 oneside 和 twoside 等。C_TE_X 会将这些选项传给标准文档类¹⁴。

5.2 章节标题风格

heading ☆

heading = (true|false)

New: 2014-03-08

本选项只能在调用 ctex.sty 时作为宏包选项使用。

C_TE_X 宏集提供了一套用于修改文档章节标题格式的接口。该选项用于选择是否启用该功能。详细的设置方法请参见第 6.3 节和第 7 节。

C_TE_X 宏集提供的四个文档类总是启用该功能。如果在 ctex.sty 下启用该选项，将会检查当前是否使用 L_AT_EX 标准文档类。若然，则该选项将会使得 ctex.sty 宏包的行为和 C_TE_X 宏集提供的四个中文文档类完全一致；若不然，则会根据 \chapter 是否有定义来使用 ctexbook 或者 ctexart 的标题设置。

sub3section ☆

sub4section ☆

修改 \paragraph 和 \subparagraph 的格式。

默认情况下，\paragraph 和 \subparagraph 会将标题与随后的正文排版在同一个段落。启用 sub3section 会将 \paragraph 的格式修改为类似 \section 的格式，并将 \subparagraph 的格式修改为原本 \paragraph 的格式。启用 sub4section 会将 \paragraph 和 \subparagraph 的格式都修改为类似 \section 的格式。

启用该选项通常还需要将计数器 secnumdepth 的值为设置为 4 或 5。

具体格式可参考 7 节中的 runin 和 afterskip 选项。

注意，上述两个选项只有在非 beamer 文档类下 heading 选项启用的时候才有意义。亦即，只有在使用了除了 ctexbeamer 的三个 C_TE_X 文档类或启用了 heading 的 ctex.sty 的时候才有意义。

¹⁴事实上，L_AT_EX 在文档类中的选项是全局设定的，除了对使用的文档类有影响外，也可能会影响到随后使用的宏包。如果这些宏包中有某些选项出现在文档类的选项列表中，那么该选项将会被自动激活。

5.3 排版方案选项

scheme ☆ `scheme = <chinese|plain>`

New: 2015-04-15

选择文章的排版方案, 预设 `chinese` 和 `plain` 两种方案。

chinese 对 `beamer` 以外的文档类, 调整默认字号为五号字, 并调整行距为 1.3; 汉化文档中的标题名字 (如“图”、“表”、“目录”和“参考文献”等, 见 6.2 节); 在 `heading = true` 的情况下¹⁵ (5.2 节), 还会将章节标题的风格修改为中文样式 (见 7 节)。

当关闭 `heading` 选项的 `ctex` 宏包与标准文档类或其衍生文档类联用时, 会载入 `indent-first` 宏包, 以实现章节标题后的段首缩进。

plain 不调整默认字号和行距, 不会汉化文档中的标题名字, 也不会将章节标题风格修改为中文样式, 同时不会调整 `\pagestyle`, 并禁用 `autoindent` 选项。事实上, 此时的 `CTEX` 宏集只提供了中文支持功能, 而不对文章版式进行任何修改。

punct `punct = <quanjiao|banjiao|kaiming|CCT|plain>`

Updated: 2014-04-11

设置标点处理格式。预定义好的格式有:

quanjiao 全角式: 所有标点占一个汉字宽度, 相邻两个标点占 1.5 汉字宽度;

banjiao 半角式: 所有标点占半个汉字宽度;

kaiming 开明式: 句末点号¹⁶用占一个汉字宽度, 标号和句内点号占半个汉字宽度;

CCT CCT 式: 所有标点符号的宽度略小于一个汉字宽度;

plain 原样 (不调整标点间距)。

space `space = <true|false|auto>`

Updated: 2014-03-08

是否在生成的 PDF 中保留汉字后面的空格。该选项仅在使用 `XYLaTeX/(pdf)LaTeX` 编译时有效。

true 总是保留汉字后的空格。此时, 用户需要自行在行尾加上 `%` 处理换行产生的空格¹⁷。

false 使用 `(pdf)LaTeX` 编译时: 总是忽略掉汉字后面的空格, 不论汉字后是什么; 使用 `XYLaTeX` 编译时, 等同于 `auto` 的效果。不建议使用该选项。

auto 根据空格后面的情况决定是否保留: 如果空格后面是汉字, 则忽略该空格, 否则保留。

例如, 使用

例 6

```
\ctexset{space=true}
汉字 分词
技术 English
```

将得到“汉字 分词 技术 English”; 使用

例 7

```
\ctexset{space=auto}
汉字 分词
技术 English
```

¹⁵使用 `CTEX` 文档类, 或者使用 `ctex` 宏包且开启该选项时。

¹⁶标点符号分为标号与点号。点号分为两类, 一共七种: 句末点号有句号、问号和叹号; 句内点号有逗号、顿号、冒号和分号。

¹⁷`LaTeX` 将单个换行视作一个空格。

则会得到“汉字分词技术 English”。

使用 `LuaATEX` 及 `upATEX` 编译的时候, 该选项无效: 汉字间的空格以及汉字与西文字符之间的空格总是有效, 不会被忽略, 但可以自动忽略掉由换行产生的空格。

`linespread` ☆ `linespread = <数值>`

New: 2014-04-23

接受一个浮点数值, 设置行距倍数。本选项的初始值与 `scheme` 有关。

`scheme = chinese` 对标准文档类初始值为 1.3, 即 1.3 倍行距。此时, 相邻两行的基线 (`\baselineskip`) 距离为 $1.3 \times 1.2 = 1.56$ 倍字体高度。对 `beamer` 不改变行距, 即使用默认的单倍行距。

`scheme = plain` `CTEX` 宏集默认不调整行距倍数, 文档中的行距由所选文档类和其他宏包或用户设置决定。

`autoindent` `autoindent = <true|false|数值|带单位的数值>`

New: 2014-03-13

在字体大小发生变化时, 是否自动调整段首缩进 (`\parindent`) 的大小。

<数值或带单位的数值> 用于设置段首缩进的长度。如果不带单位, 则默认单位是单个汉字字宽 `\ccwd`; 如果带单位, 则使用该单位。若要显式使用 `\ccwd` 为单位, 则必须在导言区进行设置。

`true` 等价于设置 `autoindent = 2` 或在导言区设置 `autoindent = 2\ccwd`。

`false` 禁用自动调整功能, 可以设置固定长度的段首缩进。如设置每段缩进 40 点:

例 8

```
\ctexset{autoindent=false}
\setlength\parindent{40pt}
```

`linestretch` ★ `linestretch = <数值或长度>`

New: 2014-03-26

`linestretch` 是一个比较特殊的选项, 它用来设置汉字之间弹性间距的弹性程度。如果有单位, 则可以在选项中直接写; 如果是数字, 单位则是汉字宽度 `\ccwd` 的倍数。

如果行宽不是汉字宽度的整数倍, 为了让段落左右两端对齐, 自然就要求伸展汉字之间的间距, 而 `linestretch` 选项就是设置每行总的允许伸行量。初始值是允许每行伸行一个汉字的宽度 `\ccwd`, 并且此宽度能根据字号变化动态调整。

过小的 `linestretch` 可能导致段落文字右侧可能参差不齐; 较大的 `linestretch` 选项则可以帮助拥有较长不可断行内容的复杂段落方便地断行, 而不会产生大量编译警告; 但很大的 `linestretch` 则会掩盖段落不良断行产生的坏盒子警告。

如果将 `linestretch` 选项的值设置为 `\maxdimen`, 则可以禁止按字号自动修改每行的允许伸长度。此时汉字间的弹性间距则固定为 `\baselineskip` 的 0.08 倍。

第 6 节 文档汉化

6.1 日期汉化

`CTEX` 宏包对显示当前日期的 `\today` 命令进行了汉化, 使之以中文的方式显示今天的日期。如本文档编译时的日期是“2020 年 5 月 2 日”。

`today` ★ `today = <small|big|old>`

该选项用来控制 `\today` 命令的输出格式:

`small` 效果为“2020 年 5 月 2 日”。使用阿拉伯数字和汉字的日期格式。

`big` 效果为“二〇二〇年五月二日”。使用全汉字的日期格式。

`old` 效果为“May 2, 2020”。使用文档原来的(英文)日期格式。

设置日期格式使用 `\ctexset` 命令完成,例如设置全汉字的日期格式:

例 9

```
\ctexset{today=big}
```

CT_EX 宏包的中文日期功能实际上是调用 `zhnumber` 宏包完成的。如果需要更多有关日期、时间的命令和更复杂的设置,可以查阅 `zhnumber` 宏包的文档。

6.2 文档标题汉化

这里主要介绍由 `scheme` 选项(5.3 节)控制的文档标题汉化功能。

设置文档标题名的示例可见例 4。下面的选项(如 `contentsname`)主要用来重新定义与选项同名的宏(如 `\contentsname`)的定义。

```
contentsname * contentsname = <名字>
```

设置目录标题名 `\contentsname`。中文默认为“目录”。

```
listfigurename * listfigurename = <名字>
```

设置插图目录标题名 `\listfigurename`。中文默认为“插图”。

```
listtablename * listtablename = <名字>
```

设置表格目录标题名 `\listtablename`。中文默认为“表格”。

```
figurename * figurename = <名字>
```

设置图片环境标题名 `\figurename`。中文默认为“图”。

```
tablename * tablename = <名字>
```

设置表格环境标题名 `\tablename`。中文默认为“表”。

```
abstractname * abstractname = <名字>
```

设置摘要 `abstract` 环境标题名 `\abstractname`。中文默认为“摘要”。注意 `book` 类没有摘要,该选项无效。

```
indexname * indexname = <名字>
```

设置索引标题名 `\indexname`。中文默认为“索引”。

```
appendixname * appendixname = <名字>
```

设置附录标题名 `\appendixname`。中文默认为“附录”。

```
bibname * bibname = <名字>
```

设置参考文献标题名。中文默认为“参考文献”。

在标准文档类中 `article` 的参考文献名使用宏 `\refname`, 而 `book` 和 `report` 使用宏 `\bibname`。本选项会根据标准文档类的不同,自动设定 `\refname` 或是 `\bibname`。因此,对于标准文档类及对应的 CT_EX 文档类可以统一地使用 `bibname` 选项来控制参考文献标题名。

对于 `beamer` 及对应的 `ctexbeamer` 来说,它们同时具有宏 `\bibname` 和宏 `\refname`。本选项仅控制其中的 `\bibname`; `\refname` 则交由 `refname` 选项控制。

`proofname` * `proofname = <名字>`

设置证明环境的名称 `\proofname`。中文默认为“证明”。

如果使用 `ctexbeamer` 文档类或者在 `beamer` 文档类下使用 `ctex` 包, 还会汉化常用定理类环境的诸如“定义”、“定理”和“引理”等名称。此时, 还有下列三个选项。

`refname` * `refname = <名字>`

设置参考文献标题名 `\refname`。中文默认为“参考文献”。

注意, 三个标准文档类(及相应的 \LaTeX 文档类)的参考文献标题名由 `bibname` 选项统一设置, 本选项仅适用于 `beamer` 及其对应的 `ctexbeamer`。在三个标准文档类(及相应的 \LaTeX 文档类)中使用 `refname` 选项会报错。

`algorithmname` * `algorithmname = <名字>`

设置算法环境标题名 `\algorithmname`。中文默认为“算法”。

`continuation` * `continuation = <名字>`

设置 `beamer` 可断页的帧在续页标题中的延续标识 `\insertcontinuationtext`。中文默认为“(续)”。

6.3 页面格式设置与汉化

页面格式设置与汉化的功能(及章节标题样式设置功能, 见第 7 节)由 `ctexheading` 宏包完成。加载该宏包时, 或者使用 \LaTeX 文档类时, 或者是使用 `ctex` 宏包并设定选项 `heading = true` 时, 相关功能被激活。此时, 整个文档的页面格式(`page style`)被设定为 `headings`, 即相当于设置了

`\pagestyle{headings}`

在页眉中显示当前章节的编号与标题。

同时, \LaTeX 宏包也会对默认的 `headings` 页面格式进行修改, 使之调用 `\CTEXthechapter`、`\CTEXthesection` 等宏来正确显示中文的章节编号。

\LaTeX 宏包的默认页面格式设置是经过汉化的 `headings`, 其基本效果如本文档所示, 只在页眉一侧显示章节编号和标题, 另一侧显示页码。

更复杂的页面格式可以通过调用 `fancyhdr`、`titleps` 等宏包来设置。 \LaTeX 宏包同时也为这些自定义页面格式的包提供了以下宏供使用:

- `\CTEXthechapter`、`\CTEXthesection` 等章节编号(见 7 节)。它们用来代替英文文档类中的 `\thechapter`、`\thesection` 等宏。
- `\leftmark`、`\rightmark`, 它们是在使用章节标题命令后, 自动设置的宏。它们实际是在与章节标题命令对应的标记命令 `\chaptermark`、`\sectionmark` 中调用 `\markright` 或 `\markboth` 生成的。

有关 \LaTeX 页面标记的含义与使用细节, 已经超出了本文档讨论的范围。可以参考 [1, Chapter 23]、[2, §4.3, §4.4] 等书籍。

这里举一个例子, 说明通过重定义 `\sectionmark`, 在 `ctexart` 文档类中的标准 `headings` 页面格式下控制页眉的方式:

例 10

```

\documentclass{ctexart}
\pagestyle{headings}
\ctexset{section={
  name={第,节},
  number=\arabic{section},
}}
\renewcommand\sectionmark[1]{%
  \markright{\CTEXifname{\CTEXthesection——}{\#1}}

\begin{document}

\section{天地玄黄}
\newpage

\section{宇宙洪荒}

\end{document}

```

在上例中,我们设置了页眉的形式是用破折号分开的节编号与节标题,即“第 1 节——天地玄黄”、“第 2 节——宇宙洪荒”。

CT_EX 宏包已经对 **fancyhdr** 宏包进行了补丁,载入 **fancyhdr** 后,其 fancy 页面格式将使用 `\CTEXthechapter` 等宏显示中文章节编号。

关于 **fancyhdr** 的具体用法可以参见其宏包手册。通常也只要像在标准的英文文档类中使用 **fancyhdr** 一样定义页眉页脚格式即可,并不需要额外的定义。

下面我则给出一个与前例类似而稍复杂的例子,展示如何在文档中设置页眉内容与页眉的格式。

例 11

```

\documentclass{ctexart}
\ctexset{section={
  name={第,节},
  number=\arabic{section},
}}
\usepackage{fancyhdr}
\fancyhf{}
\lhead{\textnormal{\kaishu\rightmark}}
\rhead{--\ \thepage\ --}
\pagestyle{fancy}
% \sectionmark 的重定义需要在 \pagestyle 之后生效
\renewcommand\sectionmark[1]{%
  \markright{\CTEXifname{\CTEXthesection——}{\#1}}

\begin{document}

\section{天地玄黄}
\newpage

\section{宇宙洪荒}

\end{document}

```

本例的页眉效果大致如下(有页眉线):

第 7 节 章节标题样式设置

CT_EX 宏集对 L^AT_EX 的标准文档类 (article、report、book) 和 beamer 进行了章节标题样式设置功能的扩充。章节标题样式设置功能 (及页面格式设置与汉化功能, 见第 6.3 节) 由 ctexheading 宏包完成。加载该宏包时, 或者使用 CT_EX 文档类时, 或者是使用 ctex 宏包并设定选项 heading = true 时, 相关功能被激活。其中, 独立使用 ctexheading 宏包时, 本节介绍各选项的默认值与指定 scheme = plain 时相同。

本节涉及的所有选项均需使用 \ctexset 命令设置。

章节标题的样式选项是分层设置的。顶层的选项是章节标题名称 (例如 section), 次一层的选项是章节标题的样式 (例如 nameformat)。章节标题名称包括 part, chapter, section, subsection, subsubsection, paragraph, subparagraph。可用的样式选项包括:

- 编号相关 (7.1 小节): numbering, name, number
- 格式相关 (7.2 小节): format, nameformat, numberformat, titleformat, aftername, aftertitle, pagestyle
- 间距、缩进相关 (7.3 小节): runin, hang, indent, beforeskip, afterskip, fixskip, break, afterindent
- 目录、附录相关 (7.4 小节): tocline, lofskip, lotskip, appendix/numbering, appendix/name, appendix/number

注意, 对 article 及其衍生的 ctexart 等文档类, 没有 chapter 级别的标题; 而对于 beamer 文档类, 这些选项控制的是由 \partpage, \sectionpage 和 \subsectionpage 产生的标题样式, 此时只有 part, section 和 subsection 这三层级别, 并且 runin, afterindent, fixskip, hang, break 和 tocline 这六个选项无效。

多层选项之间用斜线分开, 例如, part/name 选项设置 \part 标题的在数字前后的名称, 而 section/number 选项设置 \section 标题的数字类型。注意, 斜线 / 的前后不能有空格或者换行。

使用 \ctexset 设置多级选项时, 还可以在同一个上级选项下设置多个下级选项。例如, 同时设置 part 一级标题的 pagestyle 选项, chapter 一级标题的 format 与 pagestyle 选项和 section 一级标题的 name 与 number 选项:

例 12

```
\ctexset {
  part/pagestyle = empty,
  chapter = {
    format      = \raggedright,
    pagestyle = empty,
  },
  section = {
    name      = {第, 节},
    number = \chinese{section},
  }
}
```

7.1 编号相关

part/numbering	*
chapter/numbering	*
section/numbering	*
subsection/numbering	*
subsubsection/numbering	*
paragraph/numbering	*
subparagraph/numbering	*

New: 2015-06-21

numbering = true|false

控制是否对不带星号的章节标题进行编号。各级标题的默认值均为 true。

\LaTeX 标准的章节标题命令(如 `\section`)大体上完成四項工作: 输出标题内容、对标题编号(计数器增加 1)、将标题列入目录(若调用了 `hyperref` 宏包还会添加 PDF 书签)、更新页眉页脚标记。带星号的章节标题命令(如 `\section*`)只简单地输出章节标题内容,但不标题编号,不将标题列入目录或 PDF 书签,也不写入页眉页脚标记。与之不同的是,本选项仅仅是是否对不带星号的章节标题进行编号。因此,当设置本选项为 false 时,除了不对标题编号以外,其余功能与正常标题一致: 可以编入目录,并生成正确的 `hyperref` 目录超链接位置和页眉页脚标记。例如:

例 13

```
\documentclass{ctexbook}
\begin{document}
\tableofcontents
\chapter{A}
\chapter*{B}
\ctexset{chapter/numbering=false}
\chapter{C}
\end{document}
```

三章的标题分别为“第一章 A”、“B”和“C”,但在目录中则只出现“第一章 A”和“C”。

注意, 章节标题是否编号还要受到 \LaTeX 计数器 `secnumdepth` 的控制。例如, 对于 `section` 而言, 其深度为 1。因此, `section` 会被编号, 当且仅当 `secnumdepth` 不小于 1, 并且 `section/numbering` 为 true, 并且使用不带星号的章节标题命令(即 `\section`)。

如果没有特别说明, 以下将用“...”代表各级章节标题名。

.../name *

Updated: 2014-03-08

name = {<前名字>,<后名字>}

name = {<前名字>}

设置章节的名字。所谓“章节的名字”, 可以分为前后两部分, 即章节编号前后的词语, 两个词之间用一个半角逗号分开; 也可以只有一部分, 表示只有章节编号之前的名字。例如:

例 14

```
\ctexset{
  chapter/name = {第,章},
  section/name = {\S},
}
```

会使得 `\chapter` 标题使用形如“第一章”的名字, 而 `\section` 标题则使用形如“§1”的名字。该选项的默认设置见表 5。

表 5 name 选项的默认设置

标题名	scheme = chinese	scheme = plain	注
part	{第, 部分}	{\partname\space}	原 \partname 为 Part
chapter	{第, 章}	{\chaptername\space}	原 \chaptername 为 Chapter
section (beamer)	{}	{\sectionname\space}	原 \sectionname 为 \translate{Section}
section	同右	{}	
subsection (beamer)	{}	{\subsectionname\space}	原 \subsectionname 为 \translate{Subsection}
subsection	同右	{}	
subsubsection	同右	{}	
paragraph	同右	{}	
subparagraph	同右	{}	

.../number * number = {\数字输出命令}

设置章节编号的数字输出格式。⟨数字输出命令⟩ 通常是对应章节编号计数器的输出命令, 如 \thesection 或 \chinese{chapter} 之类。例如:

例 15

```
\ctexset{
  section/number = \Roman{section}
}
```

将会使 \section 的编号变为大写罗马数字(如 I、II 等)。

number 选项定义的同时将控制对章节计数器的交叉引用。在引用计数器时, 记录在 L^AT_EX 辅助文件中的是 number 选项的定义。

但是, number 选项不会影响计数器本身的输出。即设置 section/number 不会影响 \thesection 的定义(但该选项会影响 \CTEXthesection 的定义, 见后)。该选项的默认设置见表 6。

表 6 number 选项的默认设置

标题名	scheme = chinese	scheme = plain	原 \the⟨标题⟩ 等价定义
part (beamer)	\chinese{part}	\insertromanpartnumber	意义为 \Roman{part}
part	\chinese{part}	\thepart	\Roman{part}
chapter	\chinese{chapter}	\thechapter	\arabic{chapter}
section (beamer)	同右	\insertsectionnumber	意义为 \arabic{section}
section	同右	\thesection	\arabic{section}
subsection (beamer)	\arabic{section}.	\insertsubsectionnumber	意义为 \arabic{subsection}
	\arabic{subsection}		
subsection	同右	\thesubsection	\thesection.\arabic{subsection}
subsubsection	同右	\thesubsubsection	\thesubsection.\arabic{subsubsection}
paragraph	同右	\theparagraph	\thesubsubsection.\arabic{paragraph}
subparagraph	同右	\thesubparagraph	\theparagraph.\arabic{subparagraph}

7.2 格式相关

CT_EX 宏集提供了 numberformat, nameformat, titleformat, format 这几个选项用来控制章节标题的格式。它们的作用范围如图 1 所示。具体用法见下文。

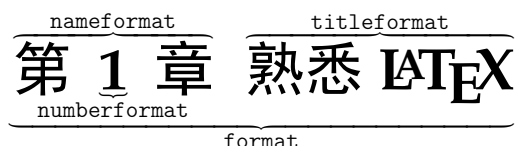


图 1 numberformat, nameformat, titleformat, format 几个选项的作用范围示意

```
.../format * format = {<格式命令>}
.../format+ * format+= {<格式命令>}
```

Updated: 2020-04-22

`format` 选项用于控制章节标题的全局格式,作用域为章节名字和随后的标题内容。可以用于控制章节标题的对齐方式、整体字体字号等格式。带加号的 `format+` 选项用于在已有的格式命令后附加内容。

`format` 选项的最后一个格式命令可以带有一个参数。这一参数用于接受章节名字、编号和标题内容(以及由 `nameformat`, `numberformat`, `aftername`, `titleformat`, `aftertitle`, `indent` 及 `hang` 选项设定的,应用于这些内容之上的格式),以实现特殊效果。

例如,设置章标题为无衬线字体左对齐、为节标题增加无衬线字体设置、为小节标题加框(`\fbox` 命令本身需带一个参数):

例 16

```
\ctexset{
  chapter/format      = \sffamily\raggedright,
  section/format      += \sffamily,
  subsection/format += \fbox,
}
```

`format` 选项的默认设置见表 7。

表 7 format 选项的默认设置

标题名	scheme = chinese	scheme = plain
part (article)	\Large\bfseries\centering	\raggedright*
part (beamer)	同右	\centering
part	\huge\bfseries\centering	\centering
chapter	\huge\bfseries\centering	\raggedright
section (beamer)	同右	\centering
section	\Large\bfseries\centering	\Large\bfseries
subsection (beamer)	同右	\centering
subsection	同右	\large\bfseries
subsubsection	同右	\normalsize\bfseries
paragraph	同右	\normalsize\bfseries
subparagraph	同右	\normalsize\bfseries

* 为了与 $\text{\LaTeX}2_{\epsilon}$ 的默认效果保持一致,在 `scheme = plain` 时, `part` 和 `chapter` 的 `nameformat` 和 `titleformat` 并不一样,因此没有使用 `format` 选项统一设置名字和标题的格式。

```
.../nameformat * nameformat = {<格式命令>}
.../nameformat+ * nameformat+= {<格式命令>}
```

Updated: 2015-06-30

`nameformat` 用于控制章节名字的格式，作用域为章节名字，包括编号。它一般用于章节名（包括编号）与章节标题的字体、字号等设置不一致的情形。参见下面的 `titleformat` 选项。
`nameformat+` 用于在已有的章节名字格式命令后附加内容。

`nameformat` 选项的最后一个格式命令可以带有一个参数。这一参数用于接受章节名字和编号，以实现特殊效果（见例 22）。

`nameformat` 选项的默认设置见表 8。

表 8 `nameformat` 选项的默认设置

标题名	scheme = chinese	scheme = plain
part (article)	{}	\Large\bfseries
part (beamer)	同右	\usebeamerfont{part name} \usebeamercolor[fg]{part name}
part	{}	\huge\bfseries
chapter	{}	\huge\bfseries
section (beamer)	同右	\usebeamerfont{section name} \usebeamercolor[fg]{section name}
section	同右	{}
subsection (beamer)	同右	\usebeamerfont{subsection name} \usebeamercolor[fg]{subsection name}
subsection	同右	{}
subsubsection	同右	{}
paragraph	同右	{}
subparagraph	同右	{}

```
.../numberformat * numberformat = {<格式命令>}
.../numberformat+ * numberformat+= {<格式命令>}
```

Updated: 2015-06-19

`numberformat` 选项用于控制章节编号的格式，作用域仅为编号数字本身。对各级标题默认均为空，当需要编号的格式和前后的章节名字不一样时可以使用。`numberformat+` 用于在已有的编号格式命令后附加内容。

`numberformat` 选项的最后一个格式命令可以带有一个参数。这一参数用于接受编号数字。例如，我们可以使用 `numberformat` 特别强调章标题中的数字：

例 17

```
\ctexset{
  chapter/number = \arabic{chapter},
  chapter/numberformat = \color{blue}\zihao{0}\emph,
}
```

上面的代码在 `scheme = chinese` 时可以做出类似这样的章标题效果：

第 4 章

`numberformat` 选项默认均设置为空，故章节编号默认与章节名字使用相同的格式。

```
.../titleformat * titleformat = {<格式命令>}
.../titleformat+ * titleformat+= {<格式命令>}
```

Updated: 2015-06-30

`titleformat` 选项用于控制标题内容的格式,作用域为章节标题内容。`titleformat+` 选项用于在已有的标题格式命令后附加内容。

`titleformat` 选项的最后一个格式命令可以带有一个参数。这一参数用于接受标题内容。例如,实现多行标题的居中悬挂对齐:

例 18

```
\usepackage{varwidth} %% 提供 varwidth 环境
\ctexset{
  chapter/name = {第,回},
  chapter/titleformat = \chaptertitleformat
}
\newcommand\chaptertitleformat[1]{%% 以标题内容为参数
  \begin{varwidth}[t]{.7\linewidth}#1\end{varwidth}}
.....
\chapter{情中情因情感妹妹\\错里错以错劝哥哥}
```

上面的代码可以做出类似这样的章标题效果:

第三十四回 情中情因情感妹妹 错里错以错劝哥哥

`titleformat` 选项的默认设置见表 9。

表 9 `titleformat` 选项的默认设置

标题名	scheme = chinese	scheme = plain
part (article)	{}	\huge\bfseries
part (beamer)	同右	\usebeamerfont{part title}
part	{}	\Huge\bfseries
chapter	{}	\Huge\bfseries
section (beamer)	同右	\usebeamerfont{section title}
section	同右	{}
subsection (beamer)	同右	\usebeamerfont{subsection title}
subsection	同右	{}
subsubsection	同右	{}
paragraph	同右	{}
subparagraph	同右	{}

```
.../aftername * aftername = {<代码>}
.../aftername+ * aftername+= {<代码>}
```

Updated: 2014-03-08

`aftername` 选项的参数 `<代码>` 将被插入到章节编号与其后的标题内容之间,用于控制格式变换。常用于控制章节编号与标题内容之间的距离,或者控制标题是否另起一行。`aftername+` 用于在已有的代码后附加内容。该选项的默认设置见表 10。

```
.../aftertitle * aftertitle = {<代码>}
.../aftertitle+ * aftertitle+= {<代码>}
```

New: 2015-06-19

`aftertitle` 选项的参数 `<代码>` 将被插入到章节标题内容之后。`aftertitle+` 用于在已有的代码后附加内容。该选项的默认设置见表 11。需注意, `sub3section` 或 `sub4section` 宏包选项(见 5.2 节)会影响 `aftertitle` 选项的默认值。

表 10 aftername 选项的默认设置

标题名	scheme = chinese	scheme = plain
part (article)	\quad	\par\nobreak
part (beamer)	同右	\vskip 1em \par
part	同右	\par\vskip 20pt
chapter	\quad	\par\nobreak\vskip 20pt
section (beamer)	同右	\vskip 1em \par
section	同右	\quad
subsection (beamer)	同右	\vskip 1em \par
subsection	同右	\quad
subsubsection	同右	\quad
paragraph	同右	\quad
subparagraph	同右	\quad

表 11 aftertitle 选项的默认设置

标题名	默认值
part	\par
chapter	\par
section	\@@par
subsection	\@@par
subsubsection	\@@par
paragraph	{}
(sub3section)	\@@par
(sub4section)	同上
subparagraph	{}
(sub4section)	\@@par

```
part/pagestyle      * pagestyle = {\页面格式}
```

```
chapter/pagestyle  *
```

```
New: 2014-03-21
```

设置 **book/ctexbook** 或 **report/ctexrep** 文档类中, \part 与 \chapter 标题所在页的页面格式 (page style)。该选项的默认设置见表 12。

表 12 pagestyle 选项的默认设置

标题名	默认值
part (article)	无效
part	plain
chapter	plain

7.3 间距、缩进相关

```
section/runin      * runin = true|false
```

```
subsection/runin   *
```

```
subsubsection/runin * runin 选项只对 \section 级以下标题有意义, 用于确定标题与随后的正文是否排在同一段之上。该选项的默认设置见表 13。
```

```
paragraph/runin    *
```

```
subparagraph/runin *
```

```
New: 2015-06-27
```

默认情况下, \paragraph、\subparagraph 两级标题是与后面正文排在同一段的, runin 选项为 true; 但使用 sub3section 或 sub4section 宏包选项 (见 5.2 节) 后, 将对这两级标题设 runin 选项为 false, 这两级标题会改为排在不同段。

表 13 runin 选项的默认设置

标题名	默认值
part	无效
chapter	无效
section	false
subsection	false
subsubsection	false
paragraph	true
(sub3section)	false
(sub4section)	同上
subparagraph	true
(sub4section)	false

.../hang ★ hang = true|false

Updated: 2020-04-23

hang 选项用于设置是否对章节标题实施悬挂缩进(缩进的宽度为名字宽度和 indent 选项设置的宽度之和)。

注意, 当 hang = true 时, 不恰当地设置选项 aftername 的值, 可能会引发错误。这是因为当 hang = true 时, L^AT_EX 内部会构造一个 \hbox 而进入受限水平模式(restricted horizontal mode)。若在 aftername 中加入包含 \vskip 等会导致从受限水平模式切出的垂直命令(vertical command)时, 就会报错。特别地, aftername 的默认值也可能导致这种情形(见 [表 10](#))。因此, 当设置 hang = true 时, 用户必须恰当地设置选项 aftername 的值。

本选项对 beamer/ctexbeamer 文档类无效。对于 \section 级以下标题, 若设置了 runin 选项为 true, 即标题与随后正文排在同一段, hang 选项没有意义。该选项的默认设置见 [表 14](#)。

.../indent ★ indent = {<缩进间距>}

Updated: 2020-04-23

indent 选项用于设置章节标题本身的首行缩进。该选项的默认设置见 [表 15](#)。

如果 indent 的值是以 em、ex 或 \ccwd 为单位, 那么缩进间距的大小是相对于 format 中指定的字号大小。例如, 设置 \part 标题缩进三个字、\section 标题缩进 20pt:

例 19

```
\ctexset{
  part = {
    format += \raggedright,
    indent = 3\ccwd,
  },
  section = {
    format = \Large\bfseries,
    indent = 20pt,
  }
}
\part{首行缩进的标题}
\noindent 无缩进的正文。
\section{首行缩进的标题}
\noindent 无缩进的正文。
```

表 14 hang 选项的默认设置

标题名	默认值
part	false
chapter	false
section	true
subsection	true
subsubsection	true
paragraph	无意义
(sub3section)	true
(sub4section)	true
subparagraph	无意义
(sub4section)	true

表 15 indent 选项的默认设置

标题名	默认值
part	0pt
chapter	0pt
section	0pt
subsection	0pt
subsubsection	0pt
paragraph	0pt
subparagraph	\parindent
(sub3section)	0pt
(sub4section)	同上

.../beforeskip ★ beforeskip = {<弹性间距>}

Updated: 2016-05-10

beforeskip 选项用于设置章节标题前的垂直间距。该选项的默认设置见 [表 16](#)。

```
.../afterskip * afterskip = {<弹性间距>}
```

Updated: 2015-06-27

`afterskip` 选项控制章节标题与后面下方之间的距离。

对于 `\section` 级以下标题, `runin` 选项会影响 `afterskip` 选项的意义: 若 `runin` 为 `true`, 标题与随后正文排在同一段, `<弹性间距>` 给出水平间距。否则, 正文另起一段, `<弹性间距>` 给出的是垂直间距。

该选项的默认设置见表 17。注意 `sub3section` 或 `sub4section` 宏包选项(见 5.2 节)会影响 `aftertitle` 选项的默认值。

表 16 beforeskip 选项的默认设置

标题名	默认值
part (article)	4ex
part (beamer)	0pt
part	0pt plus 1fil
chapter	50pt
section (beamer)	0pt
section	3.5ex plus 1ex minus .2ex
subsection (beamer)	0pt
subsection	3.25ex plus 1ex minus .2ex
subsubsection	3.25ex plus 1ex minus .2ex
paragraph	3.25ex plus 1ex minus .2ex
subparagraph	3.25ex plus 1ex minus .2ex

表 17 afterskip 选项的默认设置

标题名	默认值
part (article)	3ex
part (beamer)	0pt
part	0pt plus 1fil
chapter	40pt
section (beamer)	0pt
section	2.3ex plus .2ex
subsection (beamer)	0pt
subsection	1.5ex plus .2ex
subsubsection	1.5ex plus .2ex
paragraph	1em
(sub3section)	1ex plus .2ex
(sub4section)	同上
subparagraph	1em
(sub4section)	1ex plus .2ex

```
.../fixskip * fixskip = true|false
```

New: 2016-06-03

默认情况下, `article`、`book` 和 `report` 类的标题与正文的距离除了由 `beforeskip` 和 `afterskip` 选项设置的垂直间距外, 还会有一些多余的间距。 `fixskip` 选项用于抑制这些多余间距。该选项默认不开启。

```
.../break * break = {<格式命令>}
.../break+ * break+= {<格式命令>}
```

New: 2016-09-19

`break` 选项用于控制章节标题与之前正文的分隔关系。一般用于设置是否在标题之前分页或者设置行间罚点。 `break+` 用于在已有的格式命令后附加内容。

例如, 若当前页剩余高度小于正文高度的一半时, 则另起一页输出 `\section` 标题:

例 20

```
\usepackage{needspace}
\ctexset{section/break = \Needspace{.5\textheight}}
```

该选项的默认设置见表 18。

```
.../afterindent * afterindent = true|false
```

New: 2015-06-27

`afterindent` 选项用于设置章节标题后首段的缩进。

`book` 和 `report` 类的 `\part` 标题被单独排在一页之上, `afterindent` 选项没有意义。对于 `\section` 级以下标题, 若设置了 `runin` 选项为 `true`, 即标题与随后正文排在同一段, `afterindent` 选项也就没有了意义。

该选项的默认设置见表 19。

表 18 break 选项的默认设置

标题名	默认值
part (article)	{}
part	\if@openright\cleardoublepage\else\clearpage\fi
chapter	同上
section	\addpenalty{\@secpenalty}
subsection	同上
subsubsection	同上
paragraph	同上
subparagraph	同上

表 19 afterindent 选项的默认设置

标题名	scheme = chinese	scheme = plain
part (article)	true	false
part	无效	无效
chapter	true	false
section	true	false
subsection	true	false
subsubsection	true	false
paragraph	true	false
subparagraph	true	false

7.4 目录、附录相关

.../tocline * tocline = {(格式定义)}

New: 2016-10-25

tocline 选项用于定义章节标题在目录文件(.toc)中的格式。(格式定义)有两个参数: 参数 #1 是 part、chapter 等名字, 参数 #2 是标题内容。该选项的默认设置见表 20。

表 20 tocline 选项的默认设置

标题名	默认值
part	\CTEXifname{\CTEXthepart\hspace{1em}}{ }#2
chapter (chinese)	\CTEXifname{\protect\numberline{\CTEXthechapter\hspace{.3em}}}{ }#2
chapter (plain)	\CTEXnumberline{#1}#2
section	\CTEXnumberline{#1}#2
subsection	同上
subsubsection	同上
paragraph	同上
subparagraph	同上

这里 \CTEXnumberline 的意义是, 若标题 #1 没有名字, 则不输出 \numberline{\CTEXthe#1} 等编号:

\CTEXifname{\protect\numberline{\csname CTExthe#1\endcsname}}{ }

其中, \CTEXifname 的定义见 7.5 小节。

chapter/lofskip * lofskip = {(弹性间距)}

chapter/lotskip * lotskip = {(弹性间距)}

New: 2016-10-01

lofskip 选项控制插图目录(.lof)中, 章之间的插图标题的距离。同样, lotskip 选项控制表格目录(.lot)中, 章之间的表格标题的距离。

目前, 这两个选项只在 chapter 标题下有定义。它们的默认值, 在 scheme 选项的不同取值下都为 10pt。

appendix/numbering ★

New: 2015-06-21

numbering = true|false

控制是否对附录章（对应 **book** 与 **report**）或附录节（对应 **article**）进行编号，用法与普通章节对应的 **numbering** 选项相同。该选项默认值为 **true**。

appendix/name ★

Updated: 2014-03-08

name = {<前名字>,<后名字>}
name = {<前名字>}

设置附录章（对应 **book** 与 **report**）或附录节（对应 **article**）的名字，用法与普通章节对应的 **name** 选项相同。

注意该选项与 **appendixname** 选项（6.2 节）在意义上有些重叠，但不完全相同。**appendixname** 选项只用来重定义 `\appendixname`，而不管 `\appendixname` 如何使用；该选项则决定在章节标题中输出的名字，可以调用 `\appendixname` 设置。

该选项的默认设置见表 21。

表 21 appendix/name 选项的默认设置

文档类	影响命令	scheme = chinese	实际定义	scheme = plain	实际定义
article	<code>\section</code>	{}		{}	
book, report	<code>\chapter</code>	<code>\appendixname\space</code>	附录 □	<code>\appendixname\space</code>	Appendix□

appendix/number ★

number = {<数字输出命令>}

设置附录章（对 **book** 与 **report**）或附录节（对 **article**）编号的数字输出格式，用法与普通章节对应的 **number** 选项相同。

appendix/number 选项同时也会控制附录章节计数器的交叉引用。与普通章节的 **number** 选项类似，同样需要注意，该选项不会影响计数器本身的输出，即不影响 `\thesection` 或 `\thechapter` 的定义。

该选项的默认设置见表 22。

表 22 appendix/number 选项的默认设置

文档类	影响命令	默认值
article	<code>\section</code>	<code>\Alph{section}</code>
book, report	<code>\chapter</code>	<code>\Alph{chapter}</code>

7.5 有关命令

CT_EX 宏集还提供了一些命令（宏），用于存储章节标题格式，或进行一些条件判断。

\CTEXthepart

\CTEXthechapter

\CTEXthesection

\CTEXthesubsection

\CTEXthesubsubsection

\CTEXtheparagraph

\CTEXthesubparagraph

以 `\CTEXthe` 开头的这组宏给出结合了 **name** 与 **number** 选项的章节编号输出格式。例如在 **scheme = chinese** 时，默认章编号输出格式就是 `\CTEXthechapter`，形如“第一章”。

这组宏在 CT_EX 文档类中将代替 `\thechapter` 等宏的作用，在章节中引用本章节的完整编号。例如用于帮助定义自定义的目录格式、页眉格式等。

`\CTEXifname` `\CTEXifname` {<有名字时的内容>} {<无名字时的内容>}

New: 2016-09-18

`\CTEXifname` 会根据当前章节有无名字展开得到不同内容（通常是格式命令）。由于章节名字总是与编号一起出现，章节有无名字通常也表达为“章节是否编号”。在 \LaTeX 中，后者取决于以下几个方面：章节深度是否不大于计数器 `secnumdepth` 的值，章节标题是否使用不带星号的命令。在 \CTEX 宏集中，后者还取决于 `.../numbering` 是否为 `true`。

`\CTEXifname` 可用于 `format`, `titleformat`, `aftertitle`, `afterskip`, `indent` 这五个选项和 `\chapter` 标题 `beforeskip` 选项的格式设置之中。也可用于帮助定义自定义的目录格式、页眉格式等。

例如，设置章的标题有名字时左对齐，无名字时居中对齐，并且在标题后画一条横线。

例 21

```
\ctexset{
  chapter/format      = \CTEXifname{\raggedright}{\centering},
  chapter/aftertitle = \par\CTEXifname{}\hrule,
}
```

7.6 示例

我们最后举一个稍微复杂的例子，来看看上述选项的综合应用。

例 22

```
\ctexset {
  chapter = {
    beforeskip = 0pt,
    fixskip    = true,
    format     = \Huge\bfseries,
    nameformat = \rule{\linewidth}{1bp}\par\bigskip\hfill\chapternamebox,
    number     = \arabic{chapter},
    aftername  = \par\medskip,
    aftertitle = \par\bigskip\nointerlineskip\rule{\linewidth}{2bp}\par
  }
}
\newcommand\chapternamebox[1]{%
  \parbox{\ccwd}{\linespread{1}\selectfont\centering #1}}
.....
\chapter{熟悉 \LaTeX}
```

本例的设置效果大致如下：

第 1 章

熟悉 \LaTeX

第 1 章

第 8 节 实用命令

8.1 字号与间距

<code>\zihao</code>	<code>\zihao {<字号>}</code>
Updated: 2014-03-08	用于调整字号大小。其中 <字号> 的有效值共有 16 个，如表 23 所示。使用 <code>\zihao</code> 命令调整字体大小时，西文字号大小会始终和中文字号保持一致。

表 23 中文字号

<字号>	大小 (bp)	大小 (pt)	意义
0	42	42.157 49	初号
-0	36	36.135	小初号
1	26	26.097 49	一号
-1	24	24.09	小一号
2	22	22.082 49	二号
-2	18	18.067 49	小二号
3	16	16.06	三号
-3	15	15.056 24	小三号
4	14	14.052 49	四号
-4	12	12.045	小四号
5	10.5	10.539 37	五号
-5	9	9.033 74	小五号
6	7.5	7.528 12	六号
-6	6.5	6.524 37	小六号
7	5.5	5.520 61	七号
8	5	5.018 74	八号

<code>\ziju</code>	<code>\ziju {<中文字符宽度的倍数>}</code>
Updated: 2014-03-28	用于调整相邻汉字之间的间距，即（在正常中文行文中）前一个汉字的右边缘与后一个汉字的左边缘之间的距离。其中参数可以是任意浮点数值；而中文字符宽度指的是实际汉字的宽度，不包含当前字距。

这个命令会影响 `\ccwd` 的值，但不会影响英文字距。

<code>\ccwd</code>	当前汉字的字宽保存在长度寄存器 <code>\ccwd</code> 之中。汉字字宽是相邻两个汉字中心之间的距离，包含字距在内。因此修改字距会间接修改字宽。
Updated: 2014-03-27	

8.2 中文数字转换

CT_EX 宏集的中文数字转换功能实际上是调用 `zhnumber` 宏包来完成。下面只介绍一些基本的用法，更高级的用法可以查阅 `zhnumber` 宏包的文档。

$\backslash\text{chinese}$	$\backslash\text{chinese} \{ \langle \text{counter} \rangle \}$
Updated: 2016-05-01	$\backslash\text{pagenumbering} \{ \text{chinese} \}$
	$\backslash\text{chinese}$ 命令与 $\backslash\text{roman}$ 等命令的用法类似, 作用在一个 \LaTeX 计数器上, 将计数器的值以中文数字的形式输出。
$\backslash\text{zhnumber}$	$\backslash\text{zhnumber} \{ \langle \text{number} \rangle \}$
New: 2014-03-08	以中文格式输出数字。这里的数字可以是整数、小数和分数。
$\backslash\text{zhdigits}$	$\backslash\text{zhdigits} \{ \langle \text{number} \rangle \}$
New: 2014-03-08	将阿拉伯数字转换为中文数字串。
$\backslash\text{CTEXnumber}$	$\backslash\text{CTEXnumber} \backslash \langle \text{macro} \rangle \{ \langle \text{number} \rangle \}$
	$\backslash \langle \text{macro} \rangle$ 必须是一个 \TeX 宏, 不需预先定义。 $\backslash\text{CTEXnumber}$ 通过 $\backslash\text{zhnumber}$ 将 $\langle \text{number} \rangle$ 转为中文数字, 最后将结果存储在 $\backslash \langle \text{macro} \rangle$ 里。对 $\backslash \langle \text{macro} \rangle$ 的定义是局部的, 将它展开一次就可以得到转换结果。
	一般来说, 并不需要使用 $\backslash\text{CTEXnumber}$, 直接使用 $\backslash\text{zhnumber}$ 即可。但是, 如果在文档中需要多次使用同一个数字 $\langle \text{number} \rangle$ 的中文形式, 就可以先用 $\backslash\text{CTEXnumber}$ 将结果保存起来备用, 而不是每次使用时都用 $\backslash\text{zhnumber}$ 现场转换一次。
$\backslash\text{CTEXdigits}$	$\backslash\text{CTEXdigits} \backslash \langle \text{macro} \rangle \{ \langle \text{number} \rangle \}$
	$\backslash\text{CTEXdigits}$ 与 $\backslash\text{CTEXnumber}$ 类似, 但其转换的结果是中文数字串, 而不是中文数字。

8.3 杂项

$\backslash\text{CTeX}$	用于显示 CTeX 标志。
-------------------------	------------------------

第 9 节 $\text{Lua}\text{\LaTeX}$ 下的中文支持方式

在 $\text{Lua}\text{\LaTeX}$ 下, CTeX 宏集依赖 $\text{LuaTeX-j}\text{\LaTeX}$ 宏包来完成中文支持。该宏包是日本 \TeX 社区的北川弘典、前田一贵、八登崇之等人开发的, 设计目的主要是在 $\text{Lua}\text{\TeX}$ 引擎下实现日本 $\text{p}\text{\TeX}$ 引擎的 (大部分) 功能。它为了兼容 $\text{p}\text{\LaTeX}$ 的使用习惯, 对 $\text{\LaTeX} 2_{\epsilon}$ 的 NFSS 作了不少修改和扩充。这对于简体中文用户来说不是必要的, 因而 CTeX 禁用了它在 \LaTeX 格式下的大部分设置, 只保留了必要的部分。同时修改了它的字体设置方式, 使得相关命令与 xeCJK 宏包大致相同。

20150420 版以后的 $\text{LuaTeX-j}\text{\LaTeX}$ 宏包开始支持竖排, 但 CTeX 暂不支持竖排。

9.1 *Lua*TeX 下替代字体的设置

AlternateFont <hr/> Updated: 2020-04-30	<pre> \setCJKfamilyfont {<family>} {<base font name>} [<base font features> , AlternateFont = { {<character range₁>} {<alternate font name₁>} , {<character range₂>} {<alternate font name₂>} {<alternate font features₂>} , }]</pre>
---	--

在设置字体族 *<family>* 的时候,同时设置该字体族在字符范围 *<character range_n>* 内,对应字形的替代字体。

CharRange <hr/> New: 2014-04-14	<pre> \setCJKfamilyfont {<family>} {<alternate font name>} [CharRange = {<character range>} , <alternate font features>]</pre>
---	--

只设置字体族 *<family>* 在字符范围 *<character range>* 内,对应字形的替代字体。

一个 `\setCJKfamilyfont` 里只能使用一次 `CharRange` 或者 `AlternateFont`,但可以将它们分开重叠使用。例如下面的方式是有效的。

例 23

```

\setCJKmainfont[AlternateFont={...}{...}, ...]{...}
\setCJKmainfont[CharRange={"4E00->"67FF,-2}, ...]{...}
\setCJKmainfont[CharRange={"6800->"9FFF}, ...]{...}
```

declarecharrange ★ <hr/> New: 2020-04-30	<pre> \ctexset { declarecharrange = { {<name₁>} {<character range₁>} , {<name₂>} {<character range₂>} , ... } }</pre>
--	---

预先声明字符范围。声明字符范围 *<name>* 之后,它的名字 *<name>* 可以用在 `AlternateFont` 和 `CharRange` 选项的 *<character range>* 之中,表示对应的字符范围。

在声明字符范围 *<name>* 的同时,还为 `\setCJKmainfont` 等字体设置命令定义了选项 *<name>*,用于设置对应字符的替代字体:

```

<name> = {<alternate font name>}
<name> = {<alternate font name>} {<alternate font features>}
```

<name> 选项可以与 `AlternateFont` 共同使用,但不能与 `CharRange` 一起使用。如果没有给 *<name>* 设置值,则等价于设置 `CharRange=<name>`,即只设置 *<name>* 对应的字符范围的替代字体。

```
clearalternatefont * \ctexset
resetalternatefont * {
    clearalternatefont = {\family1, family2, ...} ,
    resetalternatefont = {\family1, family2, ...} ,
    clearalternatefont ,
    resetalternatefont
}
```

New: 2014-04-15

清除与重置 CJK 字体族 $\langle family \rangle$ 的替换字体设置。如果没有给定值,则作用于当前 CJK 字体族。清除与重置操作总是全局的。

第 10 节 C_TE_X 宏集的配置文件

C_TE_X 宏集提供了不同的配置文件,可以通过修改配置文件来改变 C_TE_X 宏集的默认行为。

在多数情况下,并不需要修改配置文件,C_TE_X 宏集的默认设置已经能满足大多数用户的需要。不恰当地修改 C_TE_X 宏集的默认行为也可能导致同一文件在别处无法正常编译或排版效果完全不同,因此修改应该慎重。

但在一些情况下,直接修改配置文件仍是必要的,例如:

- 系统没有安装默认设置的字体文件,无法编译。
- 需要经常编译来自其他系统的中文 T_EX 文件,但对方的操作系统或默认设置与本机不同。

与 C_TE_X 宏集的源代码一样,配置文件采用 L^AT_EX3 的语法编写。

C_TE_X 宏集的配置文件随宏包其他文件一起安装在 T_EX 系统 TDS 目录树中,文件后缀是 .cfg。为了避免本地配置文件内容因 C_TE_X 宏集的更新而丢失,不要直接修改系统 TDS 目录树中的配置文件,而应该将系统自带的配置文件复制到本地的或用户私有的 TDS 目录树中修改,并运行 texhash 命令刷新文件名数据库。

例如对于 T_EX Live,系统自带的配置文件就在 T_EX Live 安装目录下的 texmf-dist/tex/latex/ctex/config/ 子目录下,可以修改它的副本,保存在本地 TDS 树的 texmf-local/tex/latex/ctex/ 目录下,或者用户 TDS 树的 ~/texmf/tex/latex/ctex/ 目录下,作为本地/用户专有的配置文件。复制配置文件后需要运行 texhash 命令使本地配置文件生效。

MiK_TE_X 的配置文件也保存在类似的目录结构中,MiK_TE_X 管理的几个 TDS 根目录可以在 MiK_TE_X Options 设置项中查看到,这里不再赘述。

除了修改本地 T_EX 系统中的配置文件,对于特定文档,也可以将修改过的配置文件保存在文档的工作目录下。此时配置文件就只对工作目录下的所有文档生效。

10.1 修改宏包默认选项

配置文件 ctexopts.cfg 可以用来修改宏包的默认选项。随系统安装的配置文件除了文件信息声明外没有实际的内容,但在注释中给出了一个简单的示例,只要取消注释就可以生效。

例 24

```
% 系统自带 ctexopts.cfg 注释中的示例语句, 固定默认字体集为 windows。
% 该设置可以用在安装了 Windows 字体的非 Windows 系统中。
\ctex_set:nn { option } { fontset = windows }
```

如上例所示, 宏包选项通常使用 L^AT_EX3 的 `\ctex_set:nn` 命令完成键值设置, 第一个参数是固定的子模块 `option`, 第二个参数中是用户定义的新的默认宏包选项。

`ctexopts.cfg` 中的设置将在 C_T_EX 宏集的开始处, 定义过宏包选项之后, `\ProcessKeysOptions` 命令之前生效。最好只使用此配置文件修改宏包默认选项。

10.2 宏包载入后的配置

配置文件 `ctex.cfg` 将在宏包的末尾被载入生效。可以用它完成任意的设置, 或是覆盖已有的定义。随系统安装的配置文件除版本信息外没有实际内容, 注意配置文件中也使用 L^AT_EX3 语法。

例 25

```
% 简单的 ctex.cfg 内容示例。
% 修改默认的页面格式设置。
\pagestyle{plain}
```

例 26

```
% 略复杂的 ctex.cfg 内容示例: 禁止段末孤字成行。
% 在使用 XeTeX 编译时, 打开 xeCJK 的 CheckSingle 选项。
\sys_if_engine_xetex:T
{
  \xeCJKsetup { CheckSingle }
}
% 在使用 LuaTeX 编译时, 设置 LuaTeX-jaf 的 jcharwidowpenalty 参数。
\sys_if_engine luatex:T
{
  \ltjsetParameter { jcharwidowpenalty = 10000 }
}
```

10.3 配置标题中文翻译

由于 C_T_EX 宏集需要同时支持 GBK 和 UTF-8 两种编码, 因此对标题的中文翻译写在两个配置文件当中: `ctex-name-gbk.cfg` 和 `ctex-name-utf8.cfg`。两个文件的设置相同, 只是编码不同。

为了同一文档在不同电脑上编译效果的一致性, 通常不建议修改默认的中文翻译。

10.4 自定义字体集

4.3 节介绍的用于 `fontset` 选项的自定义字库文件, 类似于 C_T_EX 宏集的配置文件, 也应该与其他本地配置文件一起保存在本地 TDS 目录树下, 并可以配合 `ctexopts.cfg` 等配置文件使用。

第 11 节 对旧版本的兼容性

11.1 C_T_EX 0.8a 及以前的版本

在 `ctex-kit` 项目成立之前, C_T_EX 宏包的最后一个版本是 C_T_EX 0.8a (2007/05/06)。

第 2 版未考虑对这些很早版本的兼容性。

11.2 C_TE_X 0.9–C_TE_X 1.0d

在 2009 年在 ctex-kit 项目成立后, 新增了 X_YT_EX 引擎的支持, 并增加了不少控制字体的命令和选项。

这里主要介绍新版本 C_TE_X 宏包相对 1.02d 版本(2014/06/09)的兼容性。

第 2 版的 C_TE_X 宏包已尽力保证对 1.0x 版本的兼容性, 原有为 1.0x 编写的代码, 在第 2 版的 C_TE_X 宏包下保证仍能编译, 并且在大多数情况下保持编译效果不变。

C_TE_X 宏包在 0.8a 以前的版本支持以 CCT 作为底层中文支持方式, 从 0.9 版之后即不再推荐使用, 只保留向后兼容。在 C_TE_X 宏包第 2 版中则完全不再支持 CCT。

下面这些是在旧版本 C_TE_X 宏包中存在, 而在新版本中已不建议使用的选项和命令, 在未来版本中可能会删去它们的支持。

在多数情况下它们的功能仍将保留, 但也有部分选项命令功能已失效。

<hr/> cs4size c5size <hr/>	分别相当于 zihao=-4 和 zihao=5, 过时选项。
<hr/> CCT CCTfont <hr/>	相关选项已删除。
<hr/> indent noindent <hr/>	indent 和 noindent 什么也不做, 过时选项。 在中文版式下, ctex 宏包的相关功能在与标准文档类及其衍生文档类联用时默认打开。C _T E _X 文档类的相关功能由章节标题的 afterindent 选项的值来确定。
<hr/> zhmap nozhmap <hr/>	zhmap 宏包选项增加了参数, 扩充了功能, 除了支持真假值参数外, 还支持选择 zhmCJK 作为底层中文处理宏包。(4.3 节) nozhmap 选项相当于 zhmap=false。过时选项。
<hr/> winfonts adobefonts nofonts <hr/>	宏包选项 winfonts 相当于 fontset=windows, adobefonts 相当于 fontset=adobe, nofonts 相当于 fontset=none。这几个选项是过时选项, 对于新文档, 应使用 fontset 选项设置不同字体集。 另外, 第 2 版 C _T E _X 宏包的默认字体不再是 Windows 系统字体, 而是根据检测到的操作系统选择使用 Windows、Mac 的系统字体还是 Fandol 字体(4.3 节)。
<hr/> punct nopunct <hr/>	旧版本中宏包 punct 选项没有参数, 现在可以用参数设定标点风格(5.3 节)。原有无参形式的 punct 选项相当于 punct=quanjiao。 旧版宏包中 nopunct 选项的效果大致相当于 punct=plain。过时选项, 不推荐使用。
<hr/> cap nocap <hr/>	原有的 cap 和 nocap 选项由新的 scheme 选项代替。(5.3 节) cap 选项相当于 scheme = chinese, nocap 选项相当于 scheme = plain。它们均已过时, 仅因兼容性而保留。
<hr/> space nospace <hr/>	新版本宏包 space 选项增加真假值参数。(5.3 节) nospace 选项相当于 space=false, 成为过时选项。
<hr/> fancyhdr <hr/>	新版本宏包中总是自动处理对 fancyhdr 宏包的兼容性, 而由用户自己使用 \usepackage 载入 fancyhdr 宏包。 fancyhdr 选项过时, 因兼容性保留, 功能是载入 fancyhdr 宏包。

hyperref 新版本宏包中总是自动处理对 `hyperref` 宏包的兼容性, 而由用户自己使用 `\usepackage` 载入 `hyperref` 宏包。

`hyperref` 选项过时, 因兼容性保留, 功能是在导言区末尾载入 `hyperref` 宏包。

fntef 旧版本的 `fntef` 选项用于统一 `CCTfntef` 与 `CJKfntef` 的界面, 新版本 `CTEX` 宏集不再支持 `CCT`, 也不再自动载入 `CJKfntef` 或 `xeCJKfntef` 宏包, 而仅在其末尾做适当格式调整。

`fntef` 选项过时, 因兼容性保留, 功能是根据引擎载入 `CJKfntef` (`pdfEX`) 或 `xeCJKfntef` (`XETeX`) 宏包。

\CTEXunderdot
\CTEXunderline
\CTEXunderdblline
\CTEXunderwave
\CTEXsout
\CTEXxout
CTEXfilltwosides 在调用 `fntef` 宏包选项的同时, 旧版本 `CTEX` 宏包由于需要支持 `CCT` 系统, 会将以 `\CJK` 开头的 `\CJKunderline` 等宏换名为以 `\CTEX` 开头的 `\CTEXunderline` 等宏。此功能在新版本的 `CTEX` 宏集中已失去意义。此外, 在 `pdfEX` 引擎下, 用于设置格式的 `\CJKunderdotbasesep` 等宏也被更名为 `\CTEXunderdotbasesep` 等宏。

在新版本中, 上述由 `fntef` 衍生的相关命令和环境均被移除。

\CTEXsetfont 更新当前的中文字体信息, 包括当前字距 (`\ccwd`) 和段首缩进 (`\parindent`)。一般来说, 用户无需使用这个命令。

\CTEXindent 更新 `\ccwd` 宽度后设置 `\parindent=2\ccwd`。过时命令。

\CTEXnoindent 设置 `\parindent=0pt`。过时命令。

\CTEXsetup `\CTEXsetup[⟨选项⟩]{⟨标题⟩}`
相当于设置了 `\ctexset{⟨标题⟩ = {⟨选项⟩}}`。过时命令。

\CTEXoptions `\CTEXoptions[⟨选项⟩]`
相当于设置了 `\ctexset{⟨选项⟩}`。过时命令。

\Chinese `\Chinese{⟨counter⟩}`

新版宏集中 `\chinese` 统一了旧版本中 `\chinese` 和 `\Chinese` 的功能。因此, 该命令已过时。

captiondelimiter 原为 `\CTEXoptions` 命令的选项, 用于控制 `\caption` 编号后面的标点。此选项已过时, 并在新版本的 `CTEX` 宏包中失效。

可以使用 `caption` 宏包的 `labelsep` 选项来完成同样的功能。

例 27

```
% 代替 \CTEXoptions[captiondelimiter={:}]
\usepackage{caption}
\captionsetup{labelsep=colon}
```

11.3 CT_EX 1.02c 以后的 SVN 开发版

`CTEX` 宏包在 1.02c 版本 (2011/03/11) 之后在 Google code 上的 SVN 开发版本, 内部版本号一直升到 1.11 版, 但从未正式发布。SVN 开发版在 1.02c 版本的基础上新增的功能在第 2 版中大多继承了过来, 但新增的命令与选项都不再保持兼容。

`CTEX` 宏包第 2 版不保证对未发布的 SVN 开发版兼容。

11.4 C_TE_X 2.2 之前的版本

```
part/beforeskip
chapter/beforeskip
section/beforeskip
subsection/beforeskip
subsubsection/beforeskip
paragraph/beforeskip
subparagraph/beforeskip
```

在 C_TE_X 2.2 之前的版本中, beforeskip 选项的符号还用于确定章节标题后首段的缩进。当 beforeskip 是负值时, 章节标题后的第一段按英文文档的排版习惯, 没有首行缩进, 否则保留首行缩进。

这一特性在 2.2 版和后续版本中不再保留, 相应的功能通过新的 afterindent 选项来设置。如果原先设置 beforeskip 为负值, 在新版本中需要改为正值, 并设置相应的 afterindent 选项为 false。

在 C_TE_X 2.2 之前的版本中, 对于 \section 级以下标题, afterskip 选项的符号用于确定标题与随后正文是否排在同一段。如果是正值, 则正文另起一段, 否则标题与随后正文排在同一段, afterskip 的绝对值给出水平间距。

这一特性在 2.2 版和后续版本中不再保留, 相应的功能通过新的 runin 选项来设置。如果原先设置 afterskip 为负值, 在新版本中需要改为正值, 并设置相应的 runin 选项为 true。

11.5 C_TE_X 2.4.1 和 2.4.2

```
part/fixbeforeskip
chapter/fixbeforeskip
```

这两个选项已经被删除, 相应功能由新的选项 fixskip 提供。

11.6 C_TE_X 2.5 之前的版本

C_TE_X 2.5 有一些比较大的变动。

```
UTF8
GBK
```

(pdf)L_AT_EX 格式下, 文档编码初始值统一设置成 UTF-8。因此, 仍旧使用 GBK 编码的文档, 需要在文档类或宏包选项中显式指定 GBK。

```
\CTEXunderdot
\CTEXunderline
\CTEXunderdblline
\CTEXunderwave
\CTEXsout
\CTEXxout
CTEXfilltwosides
```

不再默认载入 CJKfntef 或 xeCJKfntef 宏包, 同步移除有关命令和环境。若需使用相关宏包, 建议用户使用 \usepackage 命令主动载入。

```
fntef
```

作为兼容性保留, 会视编译引擎载入相应宏包。

```
windows
```

不再支持 Windows XP 系统, 默认要求 Windows 系统有微软雅黑字体。建议 Windows XP 系统的用户及时更新操作系统。若一定要在 Windows XP 中使用, 请使用 2.5 以前的版本。

```
windowsold
windowsnew
```

过时字库选项, 作为兼容性保留, 功能是载入 windows 字库。

```
ubuntu
```

改用思源 (Noto CJK) 和文鼎字库。该字库不再支持 pdfL_AT_EX 编译。

```
AlternateFont
```

不再支持将替代字体的可选项放在字体名之前的方括号中, 新的语法是将可选项放在字体名之后的花括号之内。

除了以上列出的选项以外, 当用户使用 C_TE_X 系列文档类, 且使用 L_AT_EX 或 upL_AT_EX 编译时, 若用户没有在文档类选项中显式指定 dvips/dvipdfmx/dvisvgm 等驱动选项, 则文档类指定默认驱动为 DVIPDFMx。

第 12 节 宏集依赖情况与手工安装方法

本节介绍 C_T_EX 宏集的依赖情况，并介绍手工编译安装的具体方法。通常用户只需参照第 2.2 节介绍的方法，使用发行版自带的包管理器安装本宏集。

C_T_EX 宏集有两个源文件：`ctex.dtx`、`ctexpunct.spa`。使用不同的编译方式时，C_T_EX 依赖的宏包略有不同。在手工安装 C_T_EX 宏集之前，请确保你的 T_EX 发行版中已经正确安装了这些宏包。C_T_EX 依赖宏包的详情叙述如下：

- `expl3`、`xparse` 和 `l3keys2e` 宏包。它们属于 `l3kernel` 和 `l3packages` 宏集。
- `indentfirst` 宏包，属于 `tools` 宏集。
- `everysel` 宏包，属于 `ms` 宏集。
- `zhnumber` 宏包。
- ➔ 以上是各种编译方式都必需的依赖项。
- `CJK` 宏集。
- `CJKpunct` 宏包。
- `xCJK2uni` 宏包。
- `zhmetrics` 宏包。
- `zhmCJK` 宏包，它还依赖
 - `iftex` 宏包。
 - `ltxcmds` 宏包。
 - `kvoptions` 宏包。
 - `kvsetkeys` 宏包。
 - `keyval` 宏包，`graphics` 宏集。
- ➔ 以上是使用 pdf_L_AT_EX 或 L_AT_EX + DVIPDFM_x 的编译方式所需要的依赖项，其中 `zhm-CJK` 是可选的。
- `xeCJK` 宏集，它还依赖
 - `xtemplate` 宏包，它属于 `l3packages` 宏集。
 - `fontspec` 宏包。
- ➔ 以上是使用 X_L_AT_EX 编译时的依赖项。
- `luatexja` 宏包，它还依赖
 - `adobemapping` 宏包。
 - `luaotfload` 宏包，它还依赖 `lualibs` 宏包。
 - `luatexbase` 宏包，它还依赖 `ctablestack` 宏包。
 - `atbegshi` 宏包。
 - `etoolbox` 宏包。
 - `iftex` 宏包。
 - `infwarerr` 宏包。
 - `ltxcmds` 宏包。
 - `pdftexcmds` 宏包。
 - `xkeyval` 宏包。
- `fontspec` 宏包。
- `xunicode-addon` 宏包，属于 `xeCJK` 宏集，它还依赖
 - `xunicode` 宏包，它还依赖
 - * `graphics` 宏集。
 - * `graphics-cfg` 宏包。
 - * `graphics-def` 宏包。
 - * `tipa` 宏包。

- ➡ 以上是使用 Lua \LaTeX 编译时的依赖项。
 - pxeveryysel 宏包, 属于 platex-tools 宏集。
 - zhmetrics-uptex 宏包。
- ➡ 以上是使用 up \LaTeX 编译时的依赖项。

出于一些原因, zhmCJK 尚未被收入 \TeX Live 和 MiK \TeX 。因此, 若你希望使用 zhmCJK 作为 \CTeX 宏集的底层中文支持方式, 那么你需要自行安装该宏包。zhmCJK 的安装较为复杂。我们建议你

1. 从 CTAN 下载 zhmCJK 宏包的 **TDS 安装包**,
2. 按目录结构将文件复制到 \TeX 发行版的本地 TDS 根目录,
3. 最后执行 texhash 刷新 \TeX 发行版的 ls-R 数据库以完成安装。

其他细节, 可参照其 **宏包手册** 中第 3 节的指导。

\CTeX 宏集已被 \TeX Live 和 MiK \TeX 收录, 若无特别理由, 我们强烈建议用户使用包管理器安装本宏集。

若要手工安装, 请遵循如下步骤:

1. 从 CTAN 下载 \CTeX 宏集的 **TDS 安装包**,
2. 按目录结构将文件复制到 \TeX 发行版的本地 TDS 根目录,
3. 最后执行 texhash 刷新 \TeX 发行版的 ls-R 数据库以完成安装。

第 13 节 开发人员

- 吴凌云 (aloft@ctex.org)
- 江疆 (gzjjgod@gmail.com)
- 王越 (yuleopen@gmail.com)
- 刘海洋 (LeoLiu.PKU@gmail.com)
- 李延瑞 (LiYanrui.m2@gmail.com)
- 陈之初 (zhichu.chen@gmail.com)
- 李清 (sobenlee@gmail.com)
- 黄晨成 (liamhuang0205@gmail.com)
- 曾祥东 (xdzeng96@gmail.com)
- 李泽平 (zepinglee@gmail.com)
- 周宇恺 (muzimuzhi@gmail.com)
- 张瑞熹 (ruixizhang42@gmail.com)

参考文献

- [1] DONALD ERVIN KNUTH. *The \TeX book, Computers & Typesetting*, volume A. Addison-Wesley, 1986
- [2] FRANK MITTELBACH and MICHEL GOOSSENS. *The \LaTeX Companion*. Tools and Techniques for Computer Typesetting. Boston: Addison-Wesley, second edition, 2004

第 14 节 代码实现

1 <@@=ctex>

宏包载入检查。

```

2 <*class|ctex>
3 \tl_const:Nx \c__ctex_version_tl
4 { \cs_if_exist_use:cF { ver@ \@currname . \@currentx } { 9999/99/99 } }
5 <*class>
6 \cs_new_eq:cN { ver@ctex. \@pkgextension } \c__ctex_version_tl
7 \cs_new_eq:cN { ver@ctexcap. \@pkgextension } \c__ctex_version_tl
8 \cs_new_eq:cN { ver@ctexsize. \@pkgextension } \c__ctex_version_tl
9 \cs_new_eq:cN { ver@ctexheading. \@pkgextension } \c__ctex_version_tl
10 </class>
11 <*ctex>
12 \msg_new:nnnn { ctex } { subpackage-loaded }
13 { Package~`#1'~can~not~be~loaded~with~`ctex'. }
14 {
15   `#1'~is~actually~a~part~of~`ctex'.\\
16   It~is~not~necessary~to~load~it~separately.
17 }
18 \@ifpackageloaded { ctexsize }
19 { \msg_error:nnn { ctex } { subpackage-loaded } { ctexsize } }
20 { \cs_new_eq:cN { ver@ctexsize. \@pkgextension } \c__ctex_version_tl }
21 \@ifpackageloaded { ctexheading }
22 { \msg_error:nnn { ctex } { subpackage-loaded } { ctexheading } }
23 { \cs_new_eq:cN { ver@ctexheading. \@pkgextension } \c__ctex_version_tl }
24 </ctex>
25 </class|ctex>
26 <*class|style>
27 \RequirePackage { xparse , l3keys2e }
28 </class|style>
29 <*class|ctex>

```

检查 expl3 和 l3keys2e 的版本。

```

30 \msg_new:nnnn { ctex } { l3-too-old }
31 { Support~package~`#1'~too~old. }
32 {
33   Please~update~an~up-to-date~version~of~the~bundles\\
34   `l3kernel'~and~`l3packages'\\
35   using~your~TeX~package~manager~or~from~CTAN.
36 }
37 \@ifpackagelater { expl3 } { 2020/04/06 } { }
38 { \msg_error:nnn { ctex } { l3-too-old } { expl3 } }
39 <*class>
40 \@ifpackagelater { l3keys2e } { 2015/12/20 } { }
41 { \msg_error:nnn { ctex } { l3-too-old } { l3keys2e } }
42 </class>

```

\c__ctex_engine_str
\c__ctex_engine_file_str

引擎检查。目前 L^AT_EX3 将 A_pT_EX 识别为 u_pT_EX。

```

43 \str_const:Nx \c__ctex_engine_str
44 { \cs_if_exist:NTF \ngostype { aptex } { \c_sys_engine_str } }
45 \msg_new:nnnn { ctex } { engine-not-supported }
46 { Engine~`#1'~is~not~yet~supported,~ctex~will~abort! }
47 { You~can~switch~to~xelatex,~lualatex,~pdflatex,~uplatex,~or~aplatex. }
48 \file_if_exist:nTF { ctex-engine- \c__ctex_engine_str .def }
49 {
50   \str_const:Nx \c__ctex_engine_file_str
51   { ctex-engine- \c__ctex_engine_str .def }
52 }
53 { \msg_critical:nnx { ctex } { engine-not-supported } { \c__ctex_engine_str } }
54 </class|ctex>
55 <*class|ctex|ctexheading|ctexsize>

```

`ctexsize` 也要载入 `fix-cm` 包解决传统 `cm` 字体字号缺失的问题。

```

<!ctexsize> 56 \RequirePackage { ctexhook , ctexpatch }
<!ctexheading> 57 \RequirePackage { fix-cm }
58 <!*ctexsize>
<!ctexheading> 59 \RequirePackage { everyysel }

```

14.1 内部函数与变量

临时变量。

```

\l__ctex_tmp_tl
\l__ctex_tmp_int
\l__ctex_tmp_box
\l__ctex_tmp_dim
60 \tl_clear_new:N \l__ctex_tmp_tl
61 \int_new:N \l__ctex_tmp_int
62 \box_new:N \l__ctex_tmp_box
63 \dim_new:N \l__ctex_tmp_dim
<!ctexheading>

```

在宏包内部使用的键值选项定义、设置命令。

```

\ctex_define_option:n
\ctex_define:n
\ctex_set:n
\ctex_set:nn
64 <!/ctexsize>
65 \cs_new_protected:Npn \ctex_define_option:n
66 { \keys_define:nn { ctex / option } }
67 <!*ctexsize>
68 \cs_new_protected:Npn \ctex_define:n
69 { \keys_define:nn { ctex } }
70 \cs_new_protected:Npn \ctex_set:n
71 { \keys_set:nn { ctex } }
72 \cs_new_protected:Npn \ctex_set:nn #1
73 { \keys_set:nn { ctex / #1 } }

```

`\ctex_file_wrapper:nnn` 设置文件操作的 `\catcode` 环境,参数 #1 是设置,#2 是文件操作,#3 是恢复。默认关闭 L^AT_EX3 语法环境,并设置 @ 的 `\catcode` 为 11。

```

74 <!/ctexsize>
75 \cs_new_protected:Npn \ctex_file_wrapper:nnn #1#2#3
76 {
77   \use:x
78   {
79     \ExplSyntaxOff
80     \char_set_catcode_letter:n { 64 }
81     #1
82     \exp_not:n {#2}
83     \bool_if:NTF \l__kernel_expl_bool
84       { \ExplSyntaxOn }
85       { \ExplSyntaxOff }
86     \char_set_catcode:nn { 64 } { \char_value_catcode:n { 64 } }
87     #3
88   }
89 }

```

输入文件。

```

\ctex_file_input:n
90 \cs_new_protected:Npn \ctex_file_input:n #1
91 { \ctex_file_wrapper:nnn { } { \file_input:n {#1} } { } }
92 <!*ctexsize>

```

输入 `scheme` 文件。先查找当前文档类下的 `(scheme)`,找不到再查找一般的文件。

```

\ctex_scheme_input:n
93 \cs_new_protected:Npn \ctex_scheme_input:n #1
94 {
95   \ctex_file_wrapper:nnn
96   { }
97   {
98     \tl_if_exist:NTF \c__ctex_class_tl
99     {
100       \file_if_exist_input:nF { ctex-scheme- #1 - \c__ctex_class_tl .def }

```

```

101         { \file_input:n { ctex-scheme- #1 .def } }
102     }
103     { \file_input:n { ctex-scheme- #1 .def } }
104 }
105 { }
106 }
107 \cs_generate_variant:Nn \ctex_scheme_input:n { o }

```

`\g__ctex_section_depth_int` 若大于 3, 则 `\paragraph` 和 `\subparagraph` 标题单独占一行; 若为 3, 则 `\paragraph` 单独占一行。

```

108 <!*beamer>
109 \int_new:N \g__ctex_section_depth_int
110 \int_gset:Nn \g__ctex_section_depth_int { 2 }
111 </*!beamer>

112 </!ctexsize>
113 </class|ctex|ctexheading|ctexsize>
114 <*class|ctex>

```

对旧版本的宏包给出错误信息。

```

115 \msg_new:nnnn { ctex } { package-too-old }
116 { Support~package~`#1'~too~old. }
117 {
118     Please~update~an~up-to-date~version~of~the~package~`#1'\
119     using~your~TeX~package~manager~or~from~CTAN.
120 }

```

`\ifctexpdf` 在 `zhmetrics` 映射文件中使用。

```

121 \sys_if_output_pdf:TF
122 { \cs_new_eq:NN \ifctexpdf \if_true: }
123 { \cs_new_eq:NN \ifctexpdf \if_false: }

```

`\ctex_if_preamble:TF` 测试是否在 $\text{\LaTeX} 2_{\epsilon}$ 的导言区。在宏包内部初始为真, 文档最开始位置再设置为假。注意, 钩子 `\ctex_after_end_preamble:n` 在 `\AtBeginDocument` 之后执行, 可以与 `\@onlypreamble` 的行为一致。

```

124 \cs_new_eq:NN \ctex_if_preamble:TF \use_i:nn
125 \ctex_after_end_preamble:n { \cs_set_eq:NN \ctex_if_preamble:TF \use_ii:nn }

```

`\ctex_set_default_ccwd:Nn` 若参数 #2 带长度单位, 则设置它为 `tl` 变量 #1 的值, 否则以 `\ccwd` 为单位。

```

126 \cs_new_protected:Npn \ctex_set_default_ccwd:Nn #1#2
127 { \tl_set:Nx #1 { \__ctex_default_ccwd_aux:n {#2} } }
128 \cs_new:Npn \__ctex_default_ccwd_aux:n #1
129 {
130     \exp_not:n {#1}
131     \exp_after:wN \__ctex_default_ccwd_aux:w
132     \dim_use:N \tex_dimexpr:D #1 pt \scan_stop: \q_stop
133 }
134 \exp_last_unbraced:NNNN
135 \cs_new:Npn \__ctex_default_ccwd_aux:w #1 { \tl_to_str:n { pt } } #2 \q_stop
136 { \tl_if_empty:nT {#2} { \ccwd } }

```

`\g__ctex_encoding_tl` 所有引擎下默认编码均设为 UTF-8, 初始值为空, `\ProcessKeysOptions` 再判断。

```

137 \tl_new:N \g__ctex_encoding_tl

```

`\g__ctex_zhmCJK_bool` 是否使用 `zhmCJK` 宏包。

```

138 \bool_new:N \g__ctex_zhmCJK_bool

```

`\l__ctex_autoindent_tl` 保存 `autoindent` 选项的值,空值表示不自动调整首行缩进。

```
139 \tl_new:N \l__ctex_autoindent_tl
```

`\ctex_if_autoindent_touched:F` 检查 `autoindent` 选项是否被用户设置。

```
140 \cs_new_eq:NN \ctex_if_autoindent_touched:F \use:n
```

`\ctex_zhmap_case:nnn` 参数 #1 是 `zhmCJK` 的内容,#2 是 `zhmetrics`。

```
141 \cs_new_eq:NN \ctex_zhmap_case:nnn \use_ii:nnn
```

`\ctex_at_end:n` 区分 `\AtEndOfClass` 和 `\AtEndOfPackage`,虽然它们的意思都是一样的。

```
<class> 142 \cs_new_protected:Npn \ctex_at_end:n { \AtEndOfClass }
<ctex> 143 \cs_new_protected:Npn \ctex_at_end:n { \AtEndOfPackage }
```

`\g__ctex_std_options_clist` 保存传递给标准文档类的选项。

```
144 <*class>
145 \clist_new:N \g__ctex_std_options_clist
146 </class>
```

对无效选项给出警告。

```
147 \msg_new:nnn { ctex } { invalid-option }
148 { Option~~\l_keys_key_tl'~is~invalid~in~current~mode. }
149 \msg_new:nnn { ctex } { invalid-value }
150 { Value~~#1'~is~invalid~for~the~key~~\l_keys_key_tl'. }
```

`\ctex_deprecated_option:nn` 对过时选项或命令给出警告。

```
\ctex_set_deprecated_option:n
\ctex_deprecated_command:Nn
```

```
151 \cs_new_protected:Npn \ctex_deprecated_option:n
152 { \msg_warning:nnn { ctex } { deprecated-option } }
153 \cs_new_protected:Npn \ctex_set_deprecated_option:n #1
154 {
155   \ctex_deprecated_option:n { Option~~#1'~is~set. }
156   \ctex_set:nn { option } {#1}
157 }
158 \cs_new_protected:Npn \ctex_deprecated_command:Nn #1#2
159 {
160   \msg_warning:nnxx { ctex } { deprecated-command }
161   { \token_to_str:N #1 } { \exp_not:n {#2} }
162 }
163 \msg_new:nnn { ctex } { deprecated-option }
164 { Option~~\l_keys_key_tl'~is~deprecated.\\ #1 }
165 \msg_new:nnn { ctex } { deprecated-command }
166 { Command~~#1'~is~deprecated.\\ #2 }

167 </class|ctex>
```

`\g__ctex_font_size_int` 0 表示修改默认字体大小为五号,1 为小四号,大于 1 则不作修改。初始值 -1 表示 `zihao` 选项未初始化,会在将来根据文档类决定初值。

```
168 <*class|ctex|ctexsize>
169 \int_new:N \g__ctex_font_size_int
170 \int_gset:Nn \g__ctex_font_size_int { -1 }
171 </class|ctex|ctexsize>
```

14.2 宏包选项

```
172 <*class|style>
173 \ctex_define_option:n
174 {
175 </class|style>
```



```

zihao 176 <*class|ctex|ctexsize>
177     zihao .choice: ,
178     zihao .value_required:n = true ,
179     zihao / 5 .code:n = { \int_gset:Nn \g__ctex_font_size_int { 0 } } ,
180     zihao / -4 .code:n = { \int_gset:Nn \g__ctex_font_size_int { 1 } } ,
181     zihao / false .code:n = { \int_gset:Nn \g__ctex_font_size_int { 2 } } ,
<ctexsize> 182 }
183 </class|ctex|ctexsize>
184 <*class|ctex>
185     c5size .code:n = { \ctex_set_deprecated_option:n { zihao = 5 } } ,
186     cs4size .code:n = { \ctex_set_deprecated_option:n { zihao = -4 } } ,
187     c5size .value_forbidden:n = true ,
188     cs4size .value_forbidden:n = true ,

```

linespread 行距初始值为标志 nan, 用于检查用户是否设置了 linespread 选项。

```

189     linespread .fp_set:N = \l__ctex_line_spread_fp ,
190     linespread .initial:n = { \c_nan_fp } ,
191     linespread .value_required:n = true ,

```

autoindent 自动调整段落的首行缩进功能。

```

192     autoindent .choice: ,
193     autoindent .default:n = { true } ,
194     autoindent / true .code:n =
195     {
196         \tl_set:Nn \l__ctex_autoindent_tl { 2 \ccwd }
197         \cs_set_eq:NN \ctex_if_autoindent_touched:F \use_none:n
198     } ,
199     autoindent / false .code:n =
200     {
201         \tl_clear:N \l__ctex_autoindent_tl
202         \cs_set_eq:NN \ctex_if_autoindent_touched:F \use_none:n
203     } ,
204     autoindent / unknown .code:n =
205     {
206         \ctex_set_default_ccwd:Nn \l__ctex_autoindent_tl {#1}
207         \cs_set_eq:NN \ctex_if_autoindent_touched:F \use_none:n
208     } ,

```

indent 仅为兼容性保留, 已过时。

```

209     indent .code:n =
210     {
211         \ctex_deprecated_option:n
212         {
213             The~functionality~has~been~removed.\\
214             It's~better~to~set~the~heading~styles~via~`afterindent'~option.
215         }
216     } ,
217     indent .value_forbidden:n = true ,
218     noindent .code:n =
219     {
220         \ctex_deprecated_option:n
221         {
222             The~functionality~has~been~removed.\\
223             It's~better~to~set~the~heading~styles~via~`afterindent'~option.
224         }
225     } ,
226     noindent .value_forbidden:n = true ,

```

GBK 文档编码, 默认为 UTF-8。
UTF8

```

227     GBK .code:n =

```

```

228     {
229         \sys_if_engine_pdftex:TF
230         { \tl_gset:Nn \g__ctex_encoding_tl { GBK } }
231         {
232             \msg_warning:nn { ctex } { invalid-option }
233             \tl_gset:Nn \g__ctex_encoding_tl { UTF8 }
234         }
235     } ,
236     UTF8 .code:n = { \tl_gset:Nn \g__ctex_encoding_tl { UTF8 } } ,
237     GBK .value_forbidden:n = true ,
238     UTF8 .value_forbidden:n = true ,

```

fontset 初始值为空。若用户未指定,则根据操作系统载入对应字体配置,可以区分 Windows、macOS 和其他。

```

239     fontset .tl_gset:N = \g__ctex_fontset_tl ,
240     nofonts .code:n = { \ctex_set_deprecated_option:n { fontset = none } } ,
241     adobe-fonts .code:n = { \ctex_set_deprecated_option:n { fontset = adobe } } ,
242     winfonts .code:n = { \ctex_set_deprecated_option:n { fontset = windows } } ,
243     nofonts .value_forbidden:n = true ,
244     winfonts .value_forbidden:n = true ,
245     adobe-fonts .value_forbidden:n = true ,

```

zhmap

```

246     zhmap .choice: ,
247     zhmap .default:n = { true } ,
248     zhmap / zhmcjk .code:n =
249     {
250         \bool_gset_true:N \g__ctex_zhmcjk_bool
251         \cs_gset_eq:NN \ctex_zhmap_case:nnn \use_i:nnn
252     } ,
253     zhmap / true .code:n =
254     {
255         \bool_gset_false:N \g__ctex_zhmcjk_bool
256         \cs_gset_eq:NN \ctex_zhmap_case:nnn \use_ii:nnn
257     } ,
258     zhmap / false .code:n =
259     {
260         \bool_gset_false:N \g__ctex_zhmcjk_bool
261         \cs_gset_eq:NN \ctex_zhmap_case:nnn \use_iii:nnn
262     } ,
263     nozhmap .code:n =
264     { \ctex_set_deprecated_option:n { zhmap = false } } ,
265     nozhmap .value_forbidden:n = true ,

```

punct 设置标点符号输出格式。

```

266     punct .tl_set:N = \l__ctex_punct_tl ,
267     punct .default:n = { quanjiao } ,
268     punct .initial:n = { quanjiao } ,
269     nopunct .code:n = \ctex_set_deprecated_option:n { punct = plain } ,
270     nopunct .value_forbidden:n = true ,

```

space

```

271     space .choices:nn =
272     { true , auto , false }
273     {
274         \exp_args:Nx \ctex_at_end:n
275         { \ctex_set:n { space = \l_keys_choice_tl } }
276     } ,
277     space .default:n = { true } ,
278     nospace .code:n = { \ctex_deprecated_option:nn { space = false } } ,
279     nospace .value_forbidden:n = true ,

```

```

heading 280 heading .bool_set:N = \l__ctex_heading_bool ,
281 </class|ctex>
282 <*class|ctex|ctexheading>

sub3section 283 <!*beamer>
sub4section 284 sub3section .code:n =
285 { \int_gset:Nn \g__ctex_section_depth_int { 3 } } ,
286 sub4section .code:n =
287 { \int_gset:Nn \g__ctex_section_depth_int { 4 } } ,
288 sub3section .value_forbidden:n = true ,
289 sub4section .value_forbidden:n = true ,
290 </!beamer>

scheme 291 scheme .tl_set:N = \l__ctex_scheme_tl ,
292 <*ctexheading>
293 scheme .default:n = { plain } ,
294 scheme .initial:n = { plain }
295 }
296 </ctexheading>
297 <!*ctexheading>
298 scheme .default:n = { chinese } ,
299 scheme .initial:n = { chinese } ,
300 </!ctexheading>

301 </class|ctex|ctexheading>
302 <*class|ctex>

cap 和 nocap 是过时选项。
cap 293 cap .code:n = { \ctex_set_deprecated_option:n { scheme = chinese } } ,
nocap 294 nocap .code:n = { \ctex_set_deprecated_option:n { scheme = plain } } ,
295 cap .value_forbidden:n = true ,
296 nocap .value_forbidden:n = true ,

```

以下三项都是过时的兼容选项,它们会载入有关宏包。

```

fntef 307 fntef .code:n =
308 {
309 \sys_if_engine_xetex:TF
310 {
311 \ctex_deprecated_option:n { `xeCJKfntef'~package~is~loaded. }
312 \ctex_at_end:n { \RequirePackage { xCJKfntef } }
313 }
314 {
315 \sys_if_engine_pdftex:TF
316 {
317 \ctex_deprecated_option:n { `CJKfntef'~package~is~loaded. }
318 \ctex_at_end:n { \RequirePackage { CJKfntef } }
319 }
320 {
321 \ctex_deprecated_option:n
322 { Furthermore,~option~`fntef'~is~invalid~in~current~mode. }
323 }
324 }
325 } ,

fancyhdr 326 fancyhdr .code:n =
327 {
328 \ctex_deprecated_option:n { `fancyhdr'~package~is~loaded. }
329 \ctex_at_end:n { \RequirePackage { fancyhdr } }
330 } ,

hyperref 331 hyperref .code:n =
332 {

```

```

333     \ctex_deprecated_option:n { `hyperref'~package~will~be~loaded. }
334     \ctex_at_end:n
335     {
336         \cs_if_exist:NF \hypersetup
337         { \cs_new_eq:NN \hypersetup \ctex_hypersetup:n }
338     }
339     \ctex_at_end_preamble:n { \RequirePackage { hyperref } }
340 },
341 }

342 </class|ctex>
343 <*class|ctex|ctexsize>

```

10pt 使 **ctex** 和 **ctexsize** 可以接受文档类的全局选项,不修改默认字体大小。在文档类下还将参数
11pt 传给标准文档类。
12pt

```

344 \tl_clear_new:N \l__ctex_tmp_tl
345 \clist_map_inline:nn
346 {
347     10pt , 11pt , 12pt ,
348     8pt , 9pt , 14pt , 17pt , 20pt , 25pt , 30pt , 36pt , 48pt , 60pt
349 }
350 {
351     \tl_put_right:Nn \l__ctex_tmp_tl
352     {
353         #1 .code:n =
354         <*!class>
355             { \int_gset:Nn \g__ctex_font_size_int { 2 } } ,
356         </!class>
357         <*class>
358             {
359                 \int_gset:Nn \g__ctex_font_size_int { 2 }
360                 \clist_gput_right:Nn \g__ctex_std_options_clist {#1}
361             } ,
362         </class>
363         #1 .value_forbidden:n = true ,
364     }
365 }
366 \exp_args:No \ctex_define_option:n { \l__ctex_tmp_tl }
367 \tl_clear:N \l__ctex_tmp_tl

```

将未知选项传给标准文档类。

```

368 <*class>
369 \ctex_define_option:n
370 {
371     unknown .code:n =
372     { \clist_gput_right:No \g__ctex_std_options_clist { \CurrentOption } }
373 }
374 </class>

```

载入选项配置文件。

```

<|ctexsize> 375 \ctex_file_input:n { ctexopts.cfg }
376 </class|ctex|ctexsize>

```

处理宏包选项。

```

377 <*class|style>
378 \ProcessKeysOptions { ctex / option }
379 </class|style>

```

pdfL^AT_EX 下,如果没有显式指定编码为 UTF8,则给出警告信息。

```

380 <*class|ctex>
381 \msg_new:nnn { ctex } { pdftex-utf8 }
382 { UTF8~will~be~used~as~the~default~encoding. }

```

```

383 \tl_if_empty:NT \g__ctex_encoding_tl
384 {
385   \sys_if_engine_pdftex:T
386     { \msg_warning:nn { ctex } { pdftex-utf8 } }
387   \tl_gset:Nn \g__ctex_encoding_tl { UTF8 }
388 }
389 </class|ctex>
390 <*class>

```

五号字使用标准文档类的 10pt 字体大小设置,小四号字则使用 12pt。

```

391 \int_case:nn { \g__ctex_font_size_int }
392 {
393   { 0 } { \clist_gput_right:Nn \g__ctex_std_options_clist { 10pt } }
394   { 1 } { \clist_gput_right:Nn \g__ctex_std_options_clist { 12pt } }
395 }

```

使用 `\PassOptionsToClass` 是为了预防可能存在的选项冲突。

```

396 <*article>
397 \tl_const:Nn \c__ctex_class_tl { article }
398 \PassOptionsToClass { \g__ctex_std_options_clist } { article }
399 \LoadClass { article }
400 </article>
401 <*book>
402 \tl_const:Nn \c__ctex_class_tl { book }
403 \PassOptionsToClass { \g__ctex_std_options_clist } { book }
404 \LoadClass { book }
405 </book>
406 <*report>
407 \tl_const:Nn \c__ctex_class_tl { report }
408 \PassOptionsToClass { \g__ctex_std_options_clist } { report }
409 \LoadClass { report }
410 </report>
411 <*beamer>
412 \tl_const:Nn \c__ctex_class_tl { beamer }
413 \PassOptionsToClass { \g__ctex_std_options_clist } { beamer }
414 \LoadClass { beamer }
415 </beamer>
416 </class>

```

14.3 特定引擎支持与设置

14.3.1 ctexbackend.cfg

对于 $\text{Xe}\text{L}\text{A}\text{T}\text{E}\text{X}$ / $\text{pdf}\text{L}\text{A}\text{T}\text{E}\text{X}$ / $\text{Lua}\text{L}\text{A}\text{T}\text{E}\text{X}$ 等默认直接输出 PDF 的编译方式,用户无需为涉及驱动的宏包指定驱动选项。对于 $\text{L}\text{A}\text{T}\text{E}\text{X}$ 和 $\text{up}\text{L}\text{A}\text{T}\text{E}\text{X}$ 等默认不直接输出 PDF 的编译方式,用户则需要指定驱动选项。

由于历史遗留问题,在使用 $\text{L}\text{A}\text{T}\text{E}\text{X}$ 或 $\text{up}\text{L}\text{A}\text{T}\text{E}\text{X}$ 等编译时,大多数涉及驱动的宏包选定的默认输出驱动都是 `Dvips`。考虑当前实际使用频率,以及考虑到 $\text{C}\text{T}\text{E}\text{X}$ 宏集对中文支持的默认方式,我们在用户使用 $\text{C}\text{T}\text{E}\text{X}$ 系列文档类时,将默认的输出驱动改为 `DVIPDFMX`。

具体来说,如果 `dvips`, `dvipdfmx`, `dvisvgm` 等驱动没有在文档类的全局选项中被明确指定,我们就在 `\@classoptionslist` 开头加入 `dvipdfmx`。

本段代码只在 `ctexart` 等文档类开头载入,不在 `ctex` 中使用。并且需要放在 `expl3` 之前载入,保证它载入正确的 backend 文件。

```

417 <*backend>
418 \begingroup
419 \expandafter\ifx\csname Umathchardef\endcsname\relax
420 \else\expandafter\endgroup\expandafter\endinput\fi

```

```

421 \ifodd
422   \expandafter\ifx\csname      pdfoutput\endcsname\relax
423   \expandafter\ifx\csname enablecjktoken\endcsname\relax 0\else 1\fi
424   \else\ifnum\pdfoutput>0 0\else 1\fi\fi\space
425   \def\x#1{%
426     \if\relax\detokenize{#1}\relax
427     \gdef\@classoptionslist{dvipdfmx}%
428     \else
429       \let\CTEX@add\@ne
430       \@tfor\x:={dvips}{dvipdfmx}{dvisvgm}\do{%
431         \expandafter\in@\expandafter{\expandafter\x,}{, #1,}%
432         \ifin@ \let\CTEX@add\tw@ \@break@tfor \fi}%
433       \ifodd\CTEX@add \gdef\@classoptionslist{dvipdfmx, #1}\fi
434     \fi}
435   \expandafter\x\expandafter{\@classoptionslist}
436 \fi
437 \endgroup
438 </backend>

```

14.3.2 ctex-engine-pdfTeX.def

`\ctex_set_zhmap:n` 设置 upTeX 字体映射,同时作用于 `\AtBeginDvi` 与 `\AtBeginShipoutFirst`。该宏对 pdfTeX 和 upTeX 均有用。`\AtBeginDvi` 直接将 `\special` 保存到盒子中, `\AtBeginShipoutFirst` 是保存到宏中,并且不展开参数。

```

439 <*pdfTeX|uptex|aptex>
440 \cs_new_protected:Npn \ctex_set_zhmap:n
441 { \tl_gput_right:Nx \g__ctex_zhmap_tl }
442 \cs_new_protected:Npn \ctex_use_zhmap:
443 { \tl_use:N \g__ctex_zhmap_tl }
444 \cs_new_protected:Npn \ctex_add_dvi_zhmap:
445 { \AtBeginDvi { \ctex_use_zhmap: } }
446 \ctex_after_end_preamble:n { \ctex_add_dvi_zhmap: }
447 \ctex_at_end_package:nn { atbegshi }
448 {
449   \AtBeginShipoutFirst { \ctex_use_zhmap: }
450   \cs_gset_eq:NN \ctex_add_dvi_zhmap: \prg_do_nothing:
451 }
452 \tl_new:N \g__ctex_zhmap_tl
453 \@onlypreamble \ctex_set_zhmap:n
454 </pdfTeX|uptex|aptex>
455 <*pdfTeX>

```

`\c__ctex_cmap_encoding_seq` 需要加上 CMap 的 CJK 字体编码。

```

456 \seq_const_from_clist:Nn \c__ctex_cmap_encoding_seq
457 { C19 , C10 , C00 , C09 , C40 , C60 }

```

`\ctex_family_cmap:nn` 在 `\DeclareFontFamily` 的 `<loading-settings>` 中给 CJK 字体族加上 CMap。

```

458 \cs_new_protected:Npn \ctex_family_cmap:nn #1#2
459 {
460   \cs_if_free:cF { #1 + #2 }
461   {
462     \seq_if_in:NnT \c__ctex_cmap_encoding_seq {#1}
463     { \tl_gput_right:cn { #1 + #2 } { \ctex_add_cmap:n {#1} } }
464   }
465 }
466 \cs_generate_variant:Nn \ctex_family_cmap:nn { x }
467 \cs_new_eq:NN \CTEX@Family@CMap \ctex_family_cmap:xn

```

`\ctex_add_cmap:n` 给 #1 编码的 CJK 字体加上 CMap。

```

468 \cs_new_protected:Npn \ctex_add_cmap:n #1

```

```

469 {
470   \cs_if_free:NF \CJK@plane
471   { \ctex_add_cmap:cn { __ctex_add_cmap_ #1 \CJK@plane : } {#1} }
472 }
473 \cs_new_protected:Npn \ctex_add_cmap:Nn #1#2
474 {
475   \cs_if_exist:NF #1 { \__ctex_save_cmap:Nn #1 {#2} }
476   #1
477 }
478 \cs_generate_variant:Nn \ctex_add_cmap:Nn { c }
479 \cs_new_protected:Npn \__ctex_save_cmap:Nn #1#2
480 {
481   \tl_set:Nx \l__ctex_tmp_tl { \str_lowercase:n {#2} \CJK@plane }
482   \tex_immediate:D \tex_pdffobj:D stream ~ file { \l__ctex_tmp_tl .cmap }
483   \cs_new_protected:Npx #1
484   {
485     \exp_not:N \tex_pdffontattr:D \exp_not:N \tex_font:D
486     { /ToUnicode ~ \int_use:N \tex_pdfflastobj:D \c_space_tl 0 ~ R }
487   }
488 }

```

`\DeclareFontFamily` 只在 pdfLaTeX 下加 CMap。如 cmap 宏包被引入,则不重复设置。

```

489 \group_begin:
490 \char_set_catcode_other:N \#
491 \sys_if_output_pdf:TF
492 {
493   \group_end:
494   \ctex_appto_cmd:NnnTF \DeclareFontFamily { \ExplSyntaxOff }
495   { \CTEX@Family@CMap {#1} {#2} }
496   {
497     \ctex_at_end_package:nn { cmap }
498     { \cs_gset_eq:NN \CTEX@Family@CMap \use_none:nn }
499   }
500   { \ctex_patch_failure:N \DeclareFontFamily }
501 }
502 { \group_end: }

```

首先检查选项,决定是否载入 zhmCJK 宏包。

```

503 \if_bool:N \g__ctex_zhmCJK_bool
504 \PassOptionsToPackage { encoding = \g__ctex_encoding_tl } { zhmCJK }
505 \RequirePackage { zhmCJK }

```

不载入 zhmCJK 宏包时直接调用 CJK 及相关宏包。

```

506 \else:
507   \str_if_eq:onTF { \g__ctex_encoding_tl } { GBK }
508   { \RequirePackage { CJK } }
509   { \RequirePackage { CJKutf8 } }
510   \RequirePackage { CJKpunct , CJKspace }

```

`\ctex_load_zhmap:nnnn` 载入 zhmetrics 的字体映射文件,同时设置 \CJKrmdefault 等。

```

511 \cs_new_protected:Npn \ctex_load_zhmap:nnnn #1#2#3#4
512 {
513   \tl_set:Nn \CJKrmdefault {#1}
514   \tl_set:Nn \CJKsfdefault {#2}
515   \tl_set:Nn \CJKttdefault {#3}
516   \ctex_set_zhmap:n { \ctex_zhmap_input:n {#4} }
517 }
518 \@onlypreamble \ctex_load_zhmap:nnnn

```

`\ctex_zhmap_input:n` 载入字体映射文件时,确认 % 和 \ 的 \catcode。

```

519 \cs_new_protected:Npn \ctex_zhmap_input:n #1

```



```

520 {
521   \ctex_file_wrapper:nnn
522   {
523     \char_set_catcode_comment:n { 37 } % %
524     \char_set_catcode_escape:n { 92 } % \
525   }
526   { \file_input:n {#1} }
527   {
528     \char_set_catcode:nn { 37 } { \char_value_catcode:n { 37 } }
529     \char_set_catcode:nn { 92 } { \char_value_catcode:n { 92 } }
530   }
531 }

532 \tl_if_exist:NF \CJKfamilydefault
533 { \tl_const:Nn \CJKfamilydefault { \CJKrmdefault } }
534 \tl_if_exist:NF \CJKrmdefault { \tl_new:N \CJKrmdefault }
535 \tl_if_exist:NF \CJKsfdefault { \tl_new:N \CJKsfdefault }
536 \tl_if_exist:NF \CJKttdefault { \tl_new:N \CJKttdefault }
537 \g@addto@macro \@rmfamilyhook { \CJKfamily { \CJKrmdefault } }
538 \g@addto@macro \@sffamilyhook { \CJKfamily { \CJKsfdefault } }
539 \g@addto@macro \@ttfamilyhook { \CJKfamily { \CJKttdefault } }
540 \g@addto@macro \@defaultfamilyhook { \CJKfamily { \CJKfamilydefault } }

```

zhmCJK 判断结束。

```
541 \fi:
```

```
\ctex_CJK_input:n
\CJK@input
```

breqn 包可能会在正文中将 `~` 的 `\catcode` 改为 12 或 13, 这将破坏 CJK 对汉字的首字节的定义 (`\CJK@loadBinding` 和 `\CJK@loadEncoding`)。因此需要确保载入 `.enc` 和 `.bdg` 文件时, `~` 的 `\catcode` 为 7。

```

542 \cs_new_protected:Npn \ctex_CJK_input:n #1
543 {
544   \ctex_file_wrapper:nnn
545   {
546     \char_set_catcode_other:n { 60 } % <
547     \char_set_catcode_math_superscript:n { 94 } % ^
548     \int_set:Nn \tex_endlinechar:D { -1 }
549   }
550   { \file_input:n {#1} }
551   {
552     \char_set_catcode:nn { 60 } { \char_value_catcode:n { 60 } }
553     \char_set_catcode:nn { 94 } { \char_value_catcode:n { 94 } }
554     \int_set:Nn \tex_endlinechar:D { \int_use:N \tex_endlinechar:D }
555   }
556 }
557 \cs_set_eq:NN \CJK@input \ctex_CJK_input:n

```

```
\ctex_plane_to_utfxvibe:Nn
\CJK@surr
```

fancyhdr 宏包的 `\nouppercase` 会将 `\uppercase` 定义为 `\relax`, 而 `\CJK@surr` 需要用它将 `\CJK@plane` 转化成大写字母, 这就造成了冲突¹⁸。我们在这里给出 `\CJK@surr` 的一个不依赖 `\uppercase` 的实现。

```

558 \if_cs_exist:N \CJK@surr
559 \cs_new_protected:Npn \ctex_plane_to_utfxvibe:Nn #1#2
560 {
561   \tl_set:Nx \l__ctex_tmp_tl {#2}
562   \int_set:Nn \l__ctex_tmp_int
563   { \exp_args:No \int_from_hex:n { \l__ctex_tmp_tl } }
564   \int_compare:nNnTF \l__ctex_tmp_int < { 256 }
565   { \tl_gset:Nx #1 { \int_to_Hex:n { \l__ctex_tmp_int } } }
566   {
567     \int_sub:Nn \l__ctex_tmp_int { 256 }
568     \tl_gset:Nx #1
569     {

```

¹⁸<https://github.com/CTeX-org/ctex-kit/issues/146>

```

570             \int_to_Hex:n
571             { \int_div_truncate:nn { \l__ctex_tmp_int } { 4 } + "D800 }
572             \int_to_Hex:n
573             { \int_mod:nn { \l__ctex_tmp_int } { 4 } + "DC }
574         }
575     }
576 }
577 \cs_set_eq:NN \CJK@surr \ctex_plane_to_utfxvibe:Nn
578 \fi:

```

`\CJKpunct` 宏包会在 `\AtBeginDocument` 的里设置标点格式为 `quanjiao`。

```

579 \AtBeginDocument
580 {
581     \str_if_eq:eeF { \l__ctex_punct_tl } { quanjiao }
582     { \punctstyle { \l__ctex_punct_tl } }
583 }

```

在导言区末尾更新 `\CJKfamilydefault`, 注意要在 `\CJK@envStart` 之前使用。

```

584 \ctex_at_end_preamble:n { \ctex_update_default_family: }

```

启用中文字符功能。GBK 编码时, 将汉字的首字节设置为活动字符, 并对这些字符初始化; UTF8 编码时, 上游宏包已经处理好。 `\CJK@makeActive` 应该先于 `ctex-name-gbk.cfg` 等文件的载入。注意 `\CJK@loadBinding` 需要调用补丁后的 `\CJK@input`。使用 `zhmCJK` 时, 此功能已经被启用。

```

585 \reverse_if:N \if_bool:N \g__ctex_zhmCJK_bool
586 \str_if_eq:onF { \g__ctex_encoding_tl } { UTF8 }
587 {
588     \CJK@makeActive
589     \CJK@loadBinding { standard }
590 }

```

在导言区结束时调用 `\CJK@envStart` 启用完整的中文功能。

`\CJK@envStart` 的定义是

```

\def\CJK@envStart#1#2#3{
  \CJK@upperReset
  \ifCJK@lowercase@
    \CJK@lowerReset
  \fi%
  \CJK@makeActive%
  \CJK@global\let\CJK@selectFamily \CJK@selFam
  \CJK@global\let\CJK@selectEnc \CJK@selEnc%
  \def\CJK@@@enc{#2}
  \ifx\CJK@@@enc \@empty
    \PackageInfo{CJK}{
      no encoding parameter given,\MessageBreak
      waiting for \protect\CJKenc\space commands}
  \else
    \CJKenc{#2}
  \fi
  \CJKfontenc{#2}{#1}
  \CJKfamily{#3}
  \def\CJK@series{\f@series}
  \def\CJK@shape{\f@shape}%
  \csname CJKhook\endcsname}

```

`\CJK@upperReset` 可能会有一定风险, 因此我们直到导言区末尾才使用 `\CJK@envStart`。这样可以避免将 CJK 环境内置入 document 环境的最里层, 最后也就不需要 `\clearpage`。 `zhmCJK` 已经提供类似功能。注意先使用 `\ctex_update_default_family:` 更新 `\CJKfamilydefault`。

```

591 \exp_args:Nx \ctex_at_end_preamble:n
592 {
593     \exp_not:N \CJK@envStart

```

```

594      { } { \g__ctex_encoding_tl } { \exp_not:N \CJKfamilydefault }
595      \exp_not:N \CJKtilde
596    }

```

zhmCJK 判断结束。

```
597 \fi:
```

\ctex_auto_ignorespaces: 保存 \CJK@@ignorespaces 的定义, 方便使用。

```
598 \cs_new_eq:NN \ctex_auto_ignorespaces: \CJK@@ignorespaces
```

\ctex_ignorespaces_case:N 设置忽略空格的方式。根据 space 选项的值重定义 \CJK@ignorespaces, 并保存起来供 \ctex_set_ignorespaces: \CJKhook 备用。

```

599 \cs_new_protected:Npn \ctex_ignorespaces_case:N #1
600 {
601   \cs_set_protected:Npn \ctex_set_ignorespaces:
602     { \cs_set_eq:NN \CJK@ignorespaces #1 }
603   \ctex_set_ignorespaces:
604 }
605 \cs_new_protected:Npn \ctex_set_ignorespaces:
606 { \cs_set_eq:NN \CJK@ignorespaces \ctex_auto_ignorespaces: }

```

\CJKhook CJK 和 CJK* 环境都会重新定义 \CJK@ignorespaces。我们在 CJK 宏包提供的 \CJKhook 里重新设置它, 让这两个环境忽略空格的方式都受 space 选项的控制。这对 zhmCJK 是必要的。

```
607 \ctex_gadd_hook:Nn \CJKhook { \ctex_set_ignorespaces: }
```

\ctex_punct_set:n 设置 CJK 族对应到实际的字体。#1 是 fontset 的名字。

```

608 \cs_new_protected:Npn \ctex_punct_set:n #1
609 {
610   \clist_map_inline:Nn \c__ctex_punct_family_clist
611   {
612     \cs_if_free:cF { c__ctex_ #1 ##1 _punct_spaces_tl }
613     {
614       \cs_set_eq:cc
615       { CJKpunct@ ##1 @spaces }
616       { c__ctex_ #1 ##1 _punct_spaces_tl }
617     }
618   }
619 }
620 \clist_const:Nn \c__ctex_punct_family_clist
621 {
622   zhsong , zhhei , zhfs , zhkai , zhli , zhyou ,
623   zhsongb , zhheil , zhheib , zhyoub , zhyahei , zhyaheib
624 }

```

\ctex_punct_map_family:nn CJK 族 #1 使用族 #2 的边界信息。

```

625 \cs_new_protected:Npn \ctex_punct_map_family:nn #1#2
626 {
627   \cs_if_free:cF { CJKpunct@ #2 @spaces }
628   { \cs_set_eq:cc { CJKpunct@ #1 @spaces } { CJKpunct@ #2 @spaces } }
629 }

```

\ctex_punct_map_bfseries:nn CJK 族 #1 的 \bfseries 使用族 #2 的边界信息。

```

630 \cs_new_protected:Npn \ctex_punct_map_bfseries:nn #1#2
631 {
632   \clist_map_inline:nn {#1}
633   {
634     \ctex_punct_map_series:nnn { ##1 } { b } {#2}

```

```

635     \ctex_punct_map_series:nnn { ##1 } { bx } {#2}
636   }
637 }
638 \cs_new_protected:Npn \ctex_punct_map_series:nnn #1#2#3
639 {
640   \CJKpunctmapfamily { C19 } {#1} {#2} { m } {#3}
641   \CJKpunctmapfamily { C19 } {#1} {#2} { it } {#3}
642   \CJKpunctmapfamily { C19 } {#1} {#2} { sl } {#3}
643   \CJKpunctmapfamily { C70 } {#1} {#2} { m } {#3}
644   \CJKpunctmapfamily { C70 } {#1} {#2} { it } {#3}
645   \CJKpunctmapfamily { C70 } {#1} {#2} { sl } {#3}
646 }

```

CJK 族 #1 的 `\itshape` 使用族 #2 的边界信息。

```

647 \cs_new_protected:Npn \ctex_punct_map_itshape:nn #1#2
648 {
649   \CJKpunctmapfamily { C19 } {#1} { m } { it } {#2}
650   \CJKpunctmapfamily { C19 } {#1} { b } { it } {#2}
651   \CJKpunctmapfamily { C19 } {#1} { bx } { it } {#2}
652   \CJKpunctmapfamily { C70 } {#1} { m } { it } {#2}
653   \CJKpunctmapfamily { C70 } {#1} { b } { it } {#2}
654   \CJKpunctmapfamily { C70 } {#1} { bx } { it } {#2}
655 }

```

定义标点的边界信息。

```

\ctex_punct_space:nn
\ctexspadef

```

```

656 \cs_new_protected:Npn \ctex_punct_space:nn #1#2
657 { \tl_const:cn { c__ctex_ #1 _punct_spaces_tl } {#2} }
658 \cs_new_eq:NN \ctexspadef \ctex_punct_space:nn

```

载入边界信息文件。

```

659 \ctex_file_input:n { ctexspa.def }

660 </pdfTeX>

```

14.3.3 ctex-engine-xetex.def

```

661 <*xetex>

662 \RequirePackage { xeCJK }
663 \exp_args:Nx \xeCJKsetup
664 {
665   LoadFandol = false ,
666   PunctStyle = \l__ctex_punct_tl
667 }

```

最新版本的 `fontspec` 默认对 `\rmfamily` 和 `\sffamily` 设置 `Ligatures=TeX`，对 `\ttfamily` 设置 `WordSpace={1,0,0}` 和 `PunctuationSpace=WordSpace`。

```

668 \@ifpackagelater { fontspec } { 2014/05/25 } { }
669 { \msg_error:nnn { ctex } { package-too-old } { fontspec } }

670 </xetex>

```

14.3.4 ctex-engine-luatex.def

```

671 <*luatex>

```

LuaTeX-ja 为了兼容 p \LaTeX 的使用习惯，对 $\text{\LaTeX}_{2\epsilon}$ 的 NFSS 作了不少修改和扩充，这对于简体中文用户来说不是必要的。我们在这里禁用它。

```

672 \msg_new:nnn { ctex } { luatexja-loaded }
673 {
674   Package~`luatexja'~can~not~be~loaded~before~`ctex'.\\
675   Loading~file~`#1'~will~abort!
676 }

```

```

677 \ifpackageloaded { luatexja }
678 { \msg_critical:nxx { ctex } { luatexja-loaded } { \g_file_curr_name_str } }
679 { \tl_const:cn { ver@ltj-latex.\@pkgextension } { 9999/99/99 } }

680 \RequirePackage { luatexja }
681 \ifpackagelater { luatexja } { 2020/04/12 } { }
682 { \msg_error:nnn { ctex } { package-too-old } { luatexja } }

683 \RequirePackage { fontspec }
684 \ifpackagelater { fontspec } { 2020/02/21 } { }
685 { \msg_error:nnn { ctex } { package-too-old } { fontspec } }

```

14.3.4.1 LuaTeX-jā 的默认设置

```
686 \ExplSyntaxOff
```

以下设置抄录自 `ltjdefs.sty`, 略有改动。

U+2460–U+24FF (Enclosed Alphanumerics) 原属于字符范围 6, 是 JAchar, 我们把它归入字符范围 3, 改成 ALchar。

```

687 \ltjdefcharrange{1}{ "80-"36F, "1E00-"1EFF}
688 \ltjdefcharrange{2}{ "370-"4FF, "1F00-"1FFF}
689 \ltjdefcharrange{3}{%
690   "2000-"243F, "2460-"24FF, "2500-"27BF, "2900-"29FF, "2B00-"2BFF}
691 \ltjdefcharrange{4}{%
692   "500-"10FF, "1200-"1DFF, "2440-"245F, "27C0-"28FF, "2A00-"2AFF,
693   "2C00-"2E7F, "4DC0-"4DFF, "A4D0-"A95F, "A980-"ABFF, "E000-"F8FF,
694   "FB00-"FE0F, "FE20-"FE2F, "FE70-"FEFF, "10000-"1AFF, "1B170-"1FOFF,
695   "1F300-"1FFFF, "2000-"206F}
696 \ltjdefcharrange{5}{ "D800-"DFFF, "E0000-"E00FF, "E01F0-"10FFFF}
697 \ltjdefcharrange{6}{%
698   "2E80-"2EFF, "3000-"30FF, "3190-"319F, "31F0-"4DBF,
699   "4E00-"9FFF, "F900-"FAFF, "FE10-"FE1F, "FE30-"FE6F, "FF00-"FFEF,
700   "1B000-"1B16F, "1F100-"1F2FF, "20000-"3FFFF, "E0100-"E01EF}
701 \ltjdefcharrange{7}{%
702   "1100-"11FF, "2F00-"2FFF, "3100-"318F, "31A0-"31EF, "A000-"A4CF,
703   "A960-"A97F, "AC00-"D7FF}
704 \ltjdefcharrange{8}{ "A7, "A8, "B0, "B1, "B4, "B6, "D7, "F7}

```

将引号、破折号等中西文公用的标点符号归入字符范围 9, 将他们设置为 JAchar。

```

705 \ltjdefcharrange{9}{%
706   "2018, "2019, "201C, "201D, "2013, "2014, "2025, "2026, "2027, "2E3A}

```

LuaTeX-jā 默认把字符范围 2 和 3 设置为 JAchar, 我们这里把它们都改成 ALchar。

```

707 \ltjsetparameter{jacharrange={-1, -2, -3, -4, -5, +6, +7, -8, +9}}
708 \directlua{for x=128,255 do luatexja.math.is_math_letters[x] = true end}

```

以下设置抄录自 `ltj-latex.sty`。

```

709 \directlua{
710   local s = kpse.find_file('ltj-kinsoku.lua', 'tex')
711   luatexja.stack.charprop_stack_table[0] = s and dofile(s) or {}
712 }
713 \ltjsetparameter{kanjiskip=\z@ plus .4pt minus .5pt,
714   xkanjiskip=.25\zw plus 1pt minus 1pt,
715   autospacing, autoxspacing, jacharrange={-1},
716   yalbaselineshift=\z@, yjabaselineshift=\z@,
717   jcharwidowpenalty=500, differentjfm=paverage
718 }
719 \ExplSyntaxOn

```

14.3.4.2 LuaTeX-jā 的补丁

```
720 <@@=-ctex_ltx>
```

在 L^AT_EX 下, LuaTeX-jā 对 fontspec、xunicode、unicode-math 和 listings 打了补丁。其中前三个是把 `\char` 换成 `\ltjalchar`, 确保字符是 ALchar 类。我们这里用 xunicode-addon 来处理 xunicode。

```

721 \RequirePackage { xunicode-addon }
722 \AtBeginUTFCommand { \CTEX@beginallalchar }
723 \AtEndUTFCommand { \CTEX@endallalchar }
724 \cs_new_protected:Npn \CTEX@beginallalchar
725 {
726   \group_begin:
727   \ctex_ltj_zero_globaldefs:
728   \ltj@allalchar
729 }
730 \cs_new_protected:Npn \CTEX@endallalchar
731 { \group_end: }

```

对 listings 的补丁是让代码环境支持 JAchar 类。

```

732 \ctex_at_end_package:n { listings }
733 { \RequirePackage { lltjp-listings } }

```

14.3.4.3 Lua 函数

直接用 token.set_lua 定义,不是传统意义上的 TeX 宏。

```

734 \group_begin:
735 \char_set_catcode_space:n { 32 }
736 \lua_now:e
737 {
738   local func = lua.get_functions_table()
739   local ltjfont = luatexja.jfont
740   local getattribute = tex.getattribute
741   local set_lua = token.set_lua
742   local scan_int, scan_arg = token.scan_int, token.scan_argument
743   local new_luafunction = luatexbase.new_luafunction

```

\ctex_ltj_add_kyenc:n 保存 jfont 的编码,用于判断。

```

744   local id = new_luafunction("ctex")
745   func[id] = ltjfont.add_kyenc_list
746   set_lua("ctex_ltj_add_kyenc:n", id, "global", "protected")

```

\ctex_ltj_is_kenc:n 判断编码是否属于 jfont。

```

747   local id = new_luafunction("ctex")
748   func[id] = ltjfont.is_kenc
749   set_lua("ctex_ltj_is_kenc:n", id, "global")

```

\ctex_ltj_patch_external_font:n 若对字体的定义完全相同,则它们有相同的 font.id。因此如果字形是由 NFSS 的替换机制定义的,它们就有相同的 font.id。print_aftl_address 函数的定义是

```

function luatexja.jfont.print_aftl_address()
  return ';'ltjaltfont' .. tostring(aftl_base):sub(8)
end

```

主要目的是,如果当前字形有替代字体,则往字形的定义中加入一些标志,确保 font.id 唯一。

```

750   local id = new_luafunction("ctex")
751   func[id] = function ()
752     local s = scan_arg()
753     local is_braced, is_quoted
754     if s:sub(1,1) == '{' and s:sub(-1)=='}' then
755       is_braced = true; s = s:sub(2,-2)
756     end
757     if s:sub(1,1) == '"' and s:sub(-1) == '"' then
758       is_quoted = true; s = s:sub(2,-2)
759     end
760     s = s .. ltjfont.print_aftl_address()
761     if is_braced then s = '{'..s..'}'
762     elseif is_quoted then s = '"'..s..'"'

```

```

763     end
764     tex.sprint(-2, s)
765 end
766 set_lua("ctex_ltj_patch_external_font:n", id, "global")

```

`\ctex_ltj_use_jfont:` 使用 `jfont`, 确保当前的 `\font` 是 `jfont`。

```

767 local id = new_luafunction("ctex")
768 func[id] = function ()
769     font.current(getattribute("ltj@curjfnt"))
770 end
771 set_lua("ctex_ltj_use_jfont:", id, "global", "protected")

```

`\ctex_ltj_set_alt_font:nnnn` #1 和 #2 分别是字符区间的首末, #3 是基础字体, #4 是替代字体。

```

772 local id = new_luafunction("ctex")
773 func[id] = function ()
774     local b = tonumber(scan_arg())
775     local e = tonumber(scan_arg())
776     local alt = scan_arg()
777     local base = scan_arg()
778     ltjfont.set_alt_font_latex(b, e, alt, base)
779 end
780 set_lua("ctex_ltj_set_alt_font:nnnn", id, "global", "protected")

```

`\ctex_ltj_clear_alt_font:n` 清除 #1 的替代字体。

```

781 local id = new_luafunction("ctex")
782 func[id] = function ()
783     local base = scan_arg()
784     ltjfont.clear_alt_font_latex(base)
785 end
786 set_lua("ctex_ltj_clear_alt_font:n", id, "global", "protected")

```

`\ctex_ltj_pickup_alt_font:nn` 定义 #1 的替代字体, #2 是字体大小。

会在内部执行 `\ltj@pickup@altfont@auxy` 和 `\ltj@pickup@altfont@copy`。

```

787 local id = new_luafunction("ctex")
788 func[id] = function ()
789     local base = scan_arg()
790     local size = scan_arg()
791     ltjfont.output_alt_font_cmd("y", base)
792     ltjfont.pickup_alt_font_a(size)
793 end
794 set_lua("ctex_ltj_pickup_alt_font:nn", id, "global", "protected")

```

`_ctex_ltj_pickup_alt_font:Nn` #1 是 `font.id`, #2 是字体名称。在 `\ltj@pickup@altfont@copy` 之中使用。

```

795 local id = new_luafunction("ctex")
796 func[id] = function ()
797     local num = scan_int()
798     local base = scan_arg()
799     ltjfont.pickup_alt_font_b(num, base)
800 end
801 set_lua("_ctex_ltj_pickup_alt_font:Nn", id, "global", "protected")

```

`_ctex_ltj_if_alt_set:nT` 判断是否存在替代字体。会设置变量 `aftl_base` 和返回 `\@firstofone` 或 `\@gobble`。

```

function luatexja.jfont.does_alt_set(bbase)
    aftl_base = alt_font_table_latex[bbase]
    tex.sprint(cat_lp, aftl_base and '\\@firstofone' or '\\@gobble')
end

```

luatexja 20200412.0 以后的定义为:


```

function luatexja.jfont.does_alt_set()
  afl_base = alt_font_table_latex[scan_arg()]
  tex.sprint(cat_lp, afl_base and '\\@firstofone' or '\\@gobble')
end

802   local id = new_luafunction("ctex")
803   func[id] = \LuaTeXjversion > 20200412.0
804     and ltjfont.does_alt_set
805     or function ()
806         local base = scan_arg()
807         ltjfont.does_alt_set(base)
808     end
809   set_lua("__ctex_ltj_if_alt_set:nT", id, "global")

```

`\ctex_ltj_zero_globaldefs:` 设置 `\globaldefs` 为 0, 避免全局设置, 应当在分组中使用。这里通过 Lua 设置, 可以不受外部 TeX 环境中的 `\globaldefs` 的影响。

```

810   local id = new_luafunction("ctex")
811   func[id] = function ()
812     tex.set("globaldefs", 0)
813   end
814   set_lua("ctex_ltj_zero_globaldefs:", id, "global", "protected")
815 }
816 \group_end:

```

14.3.4.4 字体切换方式

`\ctex_ltj_select_font:` `\CJK@family` 保存的是当前 CJK 实际的字体族名, 如果为空表示没有设置过字体。

```

\ctex_ltj_select_font:
\CJK@family
817 \cs_new_protected:Npn \ctex_ltj_select_font:
818 {
819   \group_begin: \exp_args:Nnc \group_end:
820   \cs_if_exist_use:NF { \l__ctex_ltj_current_font_tl }
821   { \tl_if_empty:NF \CJK@family { \__ctex_ltj_select_font_aux: } }
822 }
823 \tl_new:N \CJK@family
824 \tl_new:N \l__ctex_ltj_current_font_tl
825 \tl_set:Nn \l__ctex_ltj_current_font_tl
826 { \CJK@encoding / \CJK@family / \f@series / \f@shape / \f@size }

```

`__ctex_ltj_select_font_aux:` 使用 `\pickup@font` 取得字体名称前, 总需要先设置 `\font@name`。在这里将 `\f@family` 换成 CJK 字体族, 并确保编码正确。

```

827 \cs_new_protected:Npn \__ctex_ltj_select_font_aux:
828 {
829   \group_begin:
830   \tl_set_eq:NN \f@encoding \CJK@encoding
831   \tl_set_eq:NN \f@family \CJK@family
832   \__ctex_ltj_push_fontname:n { \use:c { \curr@fontshape / \f@size } }
833   \ctex_ltj_pickup_font:
834   \group_end:
835   \font@name
836   \__ctex_ltj_pop_fontname:

```

当字形未定义的时候, NFSS 就会启动替换机制(`\wrong@fontshape`)。第一次启动后, `\l__ctex_ltj_current_font_tl` 还是没有定义。为此, 我们再次选择字体, 确保它有定义和指向正确的 `font.id`。这对 AlternateFont 的设置特别重要。

```

837   \cs_if_exist:cF { \l__ctex_ltj_current_font_tl }
838   { \__ctex_ltj_select_font_aux: }
839 }
840 \cs_new_protected:Npn \__ctex_ltj_push_fontname:n #1
841 {
842   \seq_gpush:No \g__ctex_ltj_fontname_seq { \font@name }
843   \cs_gset_nopar:Npx \font@name {#1}

```

```

844 }
845 \cs_new_protected:Npn \__ctex_ltj_pop_fontname:
846 {
847   \seq_gpop:NNT \g__ctex_ltj_fontname_seq \l__ctex_ltj_tmp_tl
848   { \cs_gset_eq:NN \font@name \l__ctex_ltj_tmp_tl }
849 }
850 \seq_new:N \g__ctex_ltj_fontname_seq

```

\ctex_ltj_pickup_font: 替换 \define@newfont 内部调用的 \extract@font 和 \do@subst@correction。

```

851 \cs_new_protected:Npn \ctex_ltj_pickup_font:
852 {
853   \exp_after:wN \cs_if_exist:NF \font@name
854   {
855     \group_begin:
856     \cs_set_eq:NN \extract@font \ctex_ltj_extract_font:
857     \cs_set_eq:NN \do@subst@correction \ctex_ltj_subst_font:
858     \define@newfont
859     \group_end:
860   }
861 }
862 \cs_new_eq:NN \pickup@jfont \ctex_ltj_pickup_font:

```

\ctex_ltj_extract_font: LuaTeX-jā 的 \globaljfont 在 luatexja-core 中定义:

```

%%%%\globaljfont\CS={...;jfm=metric;...}, \globaljfont
\protected\def\jfont#1{%
  \afterassignment\ltj@@jfont
  \directlua{luatexja.jfont.jfontdefX
    (false, 'yoko', '\luatexluaescapestring{\noexpand#1})}}
\protected\def\globaljfont#1{%
  \afterassignment\ltj@@jfont
  \directlua{luatexja.jfont.jfontdefX
    (true, 'yoko', '\luatexluaescapestring{\noexpand#1})}}
\newluafunction\ltj@@jfont@inner
\directlua{
  local t = lua.get_functions_table()
  t[\the\ltj@@jfont@inner] = luatexja.jfont.jfontdefY
}
\def\ltj@@jfont{\luafunction\ltj@@jfont@inner}

```

jfontdefX 函数的作用是把 \CS 定义为其后的字体, jfontdefY 的作用是更新 JFM 和记录相关字体信息。最后的工作是:

```

tex.sprint(cat_lp, global_flag, '\protected\expandafter\def\csname ',
  (cstemp==' ') and '\space' or cstemp, '\endcsname{\ltj@cur'..
  (jfm_dir == 'yoko' and 'j' or 't') .. 'fnt', fn, '\relax}')

```

\CS 的作用就是把 \ltj@curjfont 设置为刚才定义的字体的 font.id。

```

863 \cs_new_protected_nopar:Npn \ctex_ltj_extract_font:
864 {
865   \get@external@font
866   \ctex_ltj_if_alternate_shape_exist:nT { \curr@fontshape }
867   {
868     \tl_set:Nx \external@font
869     { \exp_after:wN \__ctex_ltj_patch_external_font:w \external@font }
870   }
871   \exp_after:wN \globaljfont \font@name \external@font \scan_stop:

```

这里 \font@name 不会直接改变当前字体, 而 \DeclareFontFamily 和 \DeclareFontShape 的最后一个参数通常要使用 \font 来引用当前字体。为此, 我们在分组内启用之前定义的字体, 以便能得到正确的 \font。对字体参数的赋值总是全局的, 不会受到分组的影响。

```

872   \font@name
873   \ctex_ltj_use_jfont:
874   \use:c { \f@encoding + \f@family }

```

```

875 \use:c { \curr@fontshape }
876 }

```

`\ctex_ltj_subst_font:` `\do@subst@correction` 在设置通过 `sub` 或者 `ssub` 函数定义的字体时会用到。如果没有设置 `SlantedFont`, `fontspec` 会设置 `\itdefault` 作为 `\sldefault` 的替代字形,因而会用到这个函数。它的本来定义是:

```

\def\do@subst@correction{%
  \xdef\subst@correction{%
    \font@name
    \global\expandafter\font
    \csname \curr@fontshape/\f@size\endcsname
    \noexpand\fontname\font
    \relax}%
  \aftergroup\subst@correction
}

```

我们在这里不需要定义新字体,而是设置对应字体的命令。

```

877 \cs_new_protected_nopar:Npn \ctex_ltj_subst_font:
878 {
879   \ctex_ltj_if_alter_nate_shape_exist:nF { \curr@fontshape }
880   {
881     \group_begin:
882     \tl_set_eq:NN \CJK@family \f@family
883     \cs_if_exist:cF { \l__ctex_ltj_current_font_tl }
884     {
885       \cs_gset_protected_nopar:Npx \subst@correction
886       {
887         \cs_new_eq:NN
888         \exp_not:c { \l__ctex_ltj_current_font_tl }
889         \font@name
890       }
891       \group_insert_after:N \group_insert_after:N
892       \group_insert_after:N \subst@correction
893     }
894     \group_end:
895   }
896 }

```

`\ctex_ltj_if_alter_nate_shape_exist:nTF` 即 LuaTeX-jan 中的 `\ltj@@does@alt@set`,判断是否存在替代字体。

```

897 \prg_new_conditional:Npnn \ctex_ltj_if_alter_nate_shape_exist:n #1 { T , F , TF }
898 {
899   \__ctex_ltj_if_alt_set:nT {#1} { \prg_return_true: \use_none:n }
900   \prg_return_false:
901 }

```

`__ctex_ltj_patch_external_font:w`

```

902 \cs_new:Npn \__ctex_ltj_patch_external_font:w #1 ~ at
903 { \ctex_ltj_patch_external_font:n {#1} ~ at }

```

`\ctex_ltj_select_alter_nate_font:` 在 `\selectfont` 中更新替代字体。

```

904 \cs_new_protected:Npn \ctex_ltj_select_alter_nate_font:
905 {
906   \ctex_ltj_if_alter_nate_shape_exist:nT { \l__ctex_ltj_current_shape_tl }
907   {
908     \ctex_ltj_pickup_alt_font:nn
909     { \l__ctex_ltj_current_shape_tl } { \f@size }
910   }
911 }
912 \tl_new:N \l__ctex_ltj_current_shape_tl
913 \tl_set:Nn \l__ctex_ltj_current_shape_tl
914 { \CJK@encoding / \CJK@family / \f@series / \f@shape }

```

`\ltj@pickup@altfont@auxy` 被用在函数 `output_alt_font_cmd` 中,作用是定义替代字体。

```

915 \cs_new_protected:Npn \ltj@pickup@altfont@auxy #1
916 {
917   \cs_if_exist:cF { #1/\f@size }
918   {
919     \group_begin:
920     \use:e { \exp_not:N \split@name #1 / \f@size } \@nil
921     \__ctex_ltj_push_fontname:n { \use:c { \curr@fontshape / \f@size } }
922     \ctex_ltj_pickup_font:
923     \group_end:
924     \__ctex_ltj_pop_fontname:
925   }
926 }
```

`\ltj@pickup@altfont@copy` 被用在函数 `pickup_alt_font_a` 中。`\ltj@getjfontnumber` 的作用是将字体命令 #1 对应的 `font.id` 保存到 `\ltj@tempcntc` 中。

```

927 \cs_new_protected:Npn \ltj@pickup@altfont@copy #1#2
928 {
929   \ltj@getjfontnumber #1
930   \__ctex_ltj_pickup_alt_font:Nn \ltj@tempcntc {#2}
931 }
```

14.3.4.5 数学字体族

以下内容来自 `l1tjfont.sty`,目的是让汉字可以在数学环境中直接使用。

`\ctex_ltj_if_jfont:nTF` 参数 #1 是一个 \LaTeX 2_ϵ 编码名称或者字体命令。 \LaTeX 2_ϵ 字体命令的一般形式是:

$$\langle encoding \rangle / \langle family \rangle / \langle series \rangle / \langle shape \rangle$$

通过截取名字中的 $\langle encoding \rangle$ 来判断是否是 `jfont`。最后会设置 `\ifin@` 为对应的 `\iftrue` 或者 `\iffalse`。

```

932 \cs_new:Npn \ctex_ltj_if_jfont:nTF #1
933 {
934   \ctex_ltj_is_kenc:n { \__ctex_ltj_ltj_if_jfont:w #1 / \q_stop }
935   \ifin@ \exp_after:wN \use_i:nn \else: \exp_after:wN \use_ii:nn \fi:
936 }
937 \cs_new:Npn \__ctex_ltj_ltj_if_jfont:w #1 / #2 \q_stop
938 {#1}
```

`\ctex_ltj_if_jfont_math:NTF` #1 是一个形式为 `\M@<encoding>` 的命令,它由 `\DeclareFontEncoding` 的第三个参数来定义。

```

939 \cs_new:Npn \ctex_ltj_if_jfont_math:NTF #1
940 { \exp_after:wN \__ctex_ltj_if_jfont_math:w \token_to_str:N #1 \q_stop }
941 \group_begin:
942   \char_set_catcode_other:N M
943   \cs_new:Npn \__ctex_ltj_if_jfont_math:w #1 M #2#3 \q_stop
944   { \ctex_ltj_if_jfont:nTF {#3} }
945 \group_end:
```

`\getanddefine@fonts` 在使用的场合,`\escapechar` 已经被设置成 `-1`,使用 `\token_to_str:N` 就可以得到名字,不必使用 `\cs_to_str:N`。

```

946 \cs_new_protected:Npn \ctex_ltj_get_and_define_fonts:nN #1#2
947 {
948   \exp_args:No \ctex_ltj_if_jfont:nTF { \token_to_str:N #2 }
949   { \ctex_ltj_get_and_define_fonts_je:nN }
950   { \ctex_ltj_get_and_define_fonts_al:nN }
951   {#1} #2
952 }
953 \cs_new_eq:NN \ctex_ltj_get_and_define_fonts_al:nN \getanddefine@fonts
```

```

954 \cs_set_eq:NN \getanddefine@fonts \ctex_ltj_get_and_define_fonts:nN
955 \cs_new_protected:Npn \ctex_ltj_get_and_define_fonts_ja:nN #1#2
956 {
957   \tl_gset:Nx \font@name { \use:c { \token_to_str:N #2 / \tf@size } }
958   \ctex_ltj_pickup_font: \tl_set_eq:NN \textfont@name \font@name
959   \tl_gset:Nx \font@name { \use:c { \token_to_str:N #2 / \sf@size } }
960   \ctex_ltj_pickup_font: \tl_set_eq:NN \scriptfont@name \font@name
961   \tl_gset:Nx \font@name { \use:c { \token_to_str:N #2 / \ssf@size } }
962   \ctex_ltj_pickup_font:
963   \tl_put_right:Nx \math@fonts
964   {
965     \ltj@setpar@global
966     \ltj@@set@stackfont #1 , \textfont@name \c_colon_str { MJT }
967     \ltj@@set@stackfont #1 , \scriptfont@name \c_colon_str { MJS }
968     \ltj@@set@stackfont #1 , \font@name \c_colon_str { MJSS }
969   }
970 }

```

\use@mathgroup

在使用 `unicode-math` 宏包时, `\ctex_ltj_math_group_hook:` 将被重定义。

\ctex_ltj_use_math_group:Nn

```

971 \cs_new_protected:Npn \ctex_ltj_use_math_group:Nn #1#2
972 {
973   \mode_if_math:T
974   {
975     \math@bgroup
976     \cs_if_eq:cNF { M@ \f@encoding } #1 {#1}
977     \ctex_ltj_math_group_hook:
978     \ctex_ltj_if_jfont_math:NTF #1
979     { \jfam } { \mathgroup } #2 \scan_stop:
980     \math@egroup
981   }
982 }
983 \cs_new_eq:NN \ctex_ltj_math_group_hook: \prg_do_nothing:
984 \cs_set_eq:NN \use@mathgroup \ctex_ltj_use_math_group:Nn

```

对 `unicode-math` 的补丁主要是将 `unicode-math-table.tex` 中的数学符号设置为 `lua-texja` 中的数学字母。本段代码应放在 `\ctex_ltj_math_group_hook:` 的定义之后,避免因宏包载入顺序而造成的编译错误。

```

985 \cs_new_protected:Npn \ctex_ltj_set_math_letter:NN #1#2
986 {
987   \group_begin:
988   \cs_set_protected:Npn #1 ##1##2##3
989   { \ltjsetmathletter { ##1 } }
990   #2
991   \group_end:
992 }
993 \ctex_at_end_package:n { unicode-math }
994 {
995   \cs_if_exist:NTF \um_input_math_symbol_table:
996   {
997     \ctex_ltj_set_math_letter:NN
998     \um_sym:nnn
999     \um_input_math_symbol_table:
1000   }
1001   {
1002     \cs_set_eq:NN \use@mathgroup \ctex_ltj_use_math_group:Nn
1003     \cs_set_protected:Npn \ctex_ltj_math_group_hook:
1004     { \__um_switchto_literal: }
1005     \ctex_ltj_set_math_letter:NN
1006     \__um_sym:nnn
1007     \__um_input_math_symbol_table:
1008   }
1009 }

```

14.3.4.6 字体族的定义与使用

`\ctex_mono_jfm:n`
`\l__ctex_ltj_jfm_tl` LuaTeX-ja 中与标点格式 plain 对应的 JFM 是 mono。

```
1010 \cs_new_protected:Npn \ctex_mono_jfm:n #1
1011 {
1012   \str_if_eq:nnTF {#1} { plain }
1013     { \tl_set:Nn \l__ctex_ltj_jfm_tl { mono } }
1014     { \tl_set:Nn \l__ctex_ltj_jfm_tl {#1} }
1015 }
1016 \tl_new:N \l__ctex_ltj_jfm_tl
1017 \cs_generate_variant:Nn \ctex_mono_jfm:n { o }
1018 \ctex_mono_jfm:o { \l__ctex_punct_tl }
```

`\CJK@encoding` 在 L^AT_EX 下, LuaTeX-ja 依赖字体编码来实现特殊设置。例如上述的 `\ctex_ltj_if_jfont:nTF` 就是通过判断编码来实现的, 它在设置数学字体时会用到。所以不应该与西文共用 EU2。定义字体族 song 为 `\CJK@encoding` 的默认替换字体。下划线 _ 不在 `\nfss@catcodes` 里, 可以放心使用。

```
1019 \tl_const:Nn \CJK@encoding { LTJY3 }
1020 \DeclareFontEncoding { \CJK@encoding } { } { }
1021 \use:e
1022 {
1023   \exp_not:N \DeclareFontSubstitution
1024     { \CJK@encoding } { song } { \mddefault } { \shapedefault }
1025 }
1026 \ctex_ltj_add_kyenc:n { \CJK@encoding }
1027 \DeclareFontFamily { \CJK@encoding } { song } { }
1028 \DeclareFontShape { \CJK@encoding } { song } { \mddefault } { \shapedefault }
1029   { <-> psft:SimSun:cid=Adobe-GB1-5;jfm=\l__ctex_ltj_jfm_tl } { }
1030 \DeclareFontShape { \CJK@encoding } { song } { \bfdefault } { \shapedefault }
1031   { <-> psft:SimHei:cid=Adobe-GB1-5;jfm=\l__ctex_ltj_jfm_tl } { }
1032 \tl_const:Nn \c__ctex_ltj_math_tl { CJKmath }
1033 \DeclareSymbolFont { \c__ctex_ltj_math_tl }
1034   { \CJK@encoding } { song } { \mddefault } { \shapedefault }
1035 \SetSymbolFont { \c__ctex_ltj_math_tl } { bold }
1036   { \CJK@encoding } { song } { \bfdefault } { \shapedefault }
1037 \int_const:Nn \c__ctex_ltj_math_fam_int { \use:c { sym \c__ctex_ltj_math_tl } }
1038 \jfam \c__ctex_ltj_math_fam_int
```

这是 luatexja-fontspec 中新增的一些字体选项。

```
1039 \newfontfeature { CID } { cid = #1 }
1040 \newfontfeature { JFM } { jfm = #1 }
1041 \newfontfeature { JFM-var } { jfmvar = #1 }
```

在新版本的 fontspec 中, `__fontspec_fontname_wrap:n` 变成了私有函数。

```
1042 \keys_define:nn { fontspec-preparse-external }
1043 {
1044   NoEmbed .code:n =
1045     { \cs_set_eq:NN \__fontspec_fontname_wrap:n \__ctex_ltj_noembed_wrap:n }
1046 }
1047 \cs_new:Npn \__ctex_ltj_noembed_wrap:n #1 { psft: #1 }
```

`\ctex_ltj_set_family:nnn` 将自定义的字体族名与 fontspec 实际设置的名字对应起来。

```
1048 \cs_new_protected:Npn \ctex_ltj_set_family:nnn #1#2#3
1049 {
1050   \group_begin:
1051   \clist_clear:N \l__ctex_ltj_char_range_clist
1052   \prop_clear:N \l__ctex_ltj_alternate_prop
1053   \tl_set:Nn \l__ctex_ltj_base_CJKfamily_tl {#1}
1054   \keys_set_known:nnN { ctex_ltj / fontspec } {#2} \l__ctex_ltj_tmp_tl
1055   \clist_set:No \l__ctex_ltj_font_options_clist { \l__ctex_ltj_tmp_tl }
1056   \ctex_ltj_set_alternate_family:nnF {#1} {#3}
```

```

1057     {
1058         \prop_gput:Nnn \g__ctex_ltj_family_font_name_prop {#1} {#3}
1059         \prop_gput:Nno \g__ctex_ltj_family_font_options_prop
1060             {#1} { \l__ctex_ltj_font_options_clist }
1061         \__ctex_ltj_update_family_uid:N \l__ctex_ltj_font_options_clist
1062         \__ctex_ltj_use_global_options:N \l__ctex_ltj_font_options_clist
1063         \__ctex_ltj_gset_family_cs:nn {#1} {#3}
1064     }
1065     \group_end:
1066 }
1067 \tl_new:N \l__ctex_ltj_base_CJKfamily_tl
1068 \clist_new:N \l__ctex_ltj_font_options_clist

```

__ctex_ltj_use_global_options:N 应用默认字体选项,并总是设置 JFM 和 NFSSEncoding。

```

1069 \cs_new_protected:Npn \__ctex_ltj_use_global_options:N #1
1070 {
1071     \clist_concat:NNN #1 \g__ctex_ltj_default_features_clist #1
1072     \clist_put_left:Nx #1
1073     { NFSSEncoding = \CJK@encoding , JFM = \l__ctex_ltj_jfm_tl }
1074 }

```

\g__ctex_ltj_family_name_prop 分别保存 fontspec 设置的字体族名、字体名称和字体选项。

```

\g__ctex_ltj_family_font_name_prop
\g__ctex_ltj_family_font_options_prop
1075 \prop_new:N \g__ctex_ltj_family_name_prop
1076 \prop_new:N \g__ctex_ltj_family_font_name_prop
1077 \prop_new:N \g__ctex_ltj_family_font_options_prop

```

__ctex_ltj_check_family:n 删除重复的定义,清除替代字体的先前设置。

```

1078 \cs_new_protected:Npn \__ctex_ltj_check_family:n #1
1079 {
1080     \prop_gpop:NnNT \g__ctex_ltj_family_font_name_prop {#1} \l__ctex_ltj_tmp_tl
1081     {
1082         \cs_undefine:c { \__ctex_ltj_family_csname:n {#1} }
1083         \cs_undefine:c { \__ctex_ltj_altername_cs:n {#1} }
1084         \prop_gpop:NnNT \g__ctex_ltj_family_name_prop {#1} \l__ctex_ltj_base_family_tl
1085         {
1086             \use:c { \__ctex_ltj_altername_cs:n { clear / #1 } }
1087             \cs_undefine:c { \__ctex_ltj_altername_cs:n { clear / #1 } }
1088             \cs_undefine:c { \__ctex_ltj_altername_cs:n { reset / #1 } }
1089             \prop_gremove:Nn \g__ctex_ltj_reset_altername_prop {#1}
1090         }
1091         \msg_warning:nxxx { ctex } { redefine-family } {#1} { \l__ctex_ltj_tmp_tl }
1092     }
1093 }
1094 \tl_new:N \l__ctex_ltj_tmp_tl
1095 \msg_new:nnn { ctex } { redefine-family }
1096 { Redefining~CJKfamily~`\__ctex_ltj_msg_family_map:n {#1}'~(#2). }

```

__ctex_ltj_gset_family_cs:nn 在设置字体时,实际上并不是马上就定义。而是只保存相关参数,在通过 \CJKfamily 第一次使用时才定义。需要注意将编码改为 \CJK@encoding。

```

1097 \cs_new_protected:Npn \__ctex_ltj_gset_family_cs:nn #1#2
1098 {
1099     \cs_gset_protected:cpx { \__ctex_ltj_family_csname:n {#1} }
1100     {
1101         \group_begin:
1102         \exp_not:n { \cs_set_eq:NN \CJKfamily \use_none:n }
1103         \exp_not:n { \fontspec_gset_family:Nnn \g__ctex_ltj_fontspec_family_tl }
1104             { \exp_not:o { \l__ctex_ltj_font_options_clist } } {#2}
1105         \prop_gput:Nno \exp_not:N \g__ctex_ltj_family_name_prop {#1}
1106             { \exp_not:N \g__ctex_ltj_fontspec_family_tl }
1107         \__ctex_ltj_set_altername_family:n {#1}
1108         \group_end:
1109     }

```

```

1110 }
1111 \tl_new:N \l__ctex_ltj_base_family_tl
1112 \tl_new:N \g__ctex_ltj_fontspec_family_tl
1113 \cs_new:Npn \__ctex_ltj_family_csname:n #1 { ctex_ltj/family/#1 }
1114 \cs_new_protected:Npn \__ctex_ltj_set_alterate_family:n #1
1115 {
1116   \tl_set:Nn \l__ctex_ltj_base_CJKfamily_tl {#1}
1117   \tl_set_eq:NN \l__ctex_ltj_base_family_tl \g__ctex_ltj_fontspec_family_tl
1118   \cs_if_exist_use:c { \__ctex_ltj_alterate_cs:n { reset / #1 } }
1119   \cs_if_exist_use:c { \__ctex_ltj_alterate_cs:n {#1} }
1120 }
1121 \cs_new:Npn \__ctex_ltj_alterate_cs:n #1 { ctex_ltj/alterate_family/#1 }

```

\CJKfamily 切换字体。

```

1122 \NewDocumentCommand \CJKfamily { m }
1123 { \ctex_ltj_switch_family:x {#1} \tex_ignorespaces:D }
1124 \cs_new_protected:Npn \ctex_ltj_switch_family:n #1
1125 {
1126   \ctex_ltj_family_if_exist:nNTF {#1} \CJK@family
1127   {
1128     \tl_set:Nn \l__ctex_ltj_family_tl {#1}
1129     \selectfont
1130   }
1131   { \__ctex_ltj_family_unknown_warning:n {#1} }
1132 }
1133 \tl_new:N \l__ctex_ltj_family_tl
1134 \cs_generate_variant:Nn \ctex_ltj_switch_family:n { x }

```

\ctex_ltj_family_if_exist:nNTF 判断 CJK 字体族 #1 是否存在, 若存在则把实际族名保存到 #2 中。

```

1135 \prg_new_protected_conditional:Npnn \ctex_ltj_family_if_exist:n #1#2 { T , F , TF }
1136 {
1137   \prop_get:NnNTF \g__ctex_ltj_family_name_prop {#1} #2
1138   { \prg_return_true: }
1139   {
1140     \cs_if_exist_use:cTF { \__ctex_ltj_family_csname:n {#1} }
1141     {
1142       \tl_set_eq:NN #2 \g__ctex_ltj_fontspec_family_tl
1143       \prg_return_true:
1144     }
1145     { \prg_return_false: }
1146   }
1147 }
1148 \prg_generate_conditional_variant:Nnn \ctex_ltj_family_if_exist:n { x } { T , F , TF }

```

```

\__ctex_ltj_family_unknown_warning:n 1149 \cs_new_protected:Npn \__ctex_ltj_family_unknown_warning:n #1
1150 {
1151   \prop_if_empty:NF \g__ctex_ltj_family_font_name_prop
1152   {
1153     \seq_if_in:NnF \g__ctex_ltj_unknown_family_seq {#1}
1154     {
1155       \seq_gput_right:Nn \g__ctex_ltj_unknown_family_seq {#1}
1156       \msg_warning:nnn { ctex } { family-unknown } {#1}
1157     }
1158   }
1159 }
1160 \seq_new:N \g__ctex_ltj_unknown_family_seq
1161 \msg_new:nnn { ctex } { family-unknown }
1162 {
1163   Unknown~CJK~family~\__ctex_ltj_msg_family_map:n {#1}'~is~being~ignored.\\
1164   Try~to~use~\__ctex_ltj_msg_def_family_map:n {#1}'~to~define~it.
1165 }
1166 \cs_new:Npn \__ctex_ltj_msg_def_family_map:n #1
1167 {
1168   \str_case_e:nnF {#1}
1169   {

```



```

1170         \CJKrmdefault { \token_to_str:N \setCJKmainfont }
1171         \CJKsfdefault { \token_to_str:N \setCJKsansfont }
1172         \CJKttdefault { \token_to_str:N \setCJKmonofont }
1173     }
1174     { \token_to_str:N \setCJKfamilyfont \{ #1 \} }
1175     [...] \{...\}
1176 }
1177 \cs_new:Npn \__ctex_ltj_msg_family_map:n #1
1178 {
1179     \str_case:e:nnF {#1}
1180     {
1181         \CJKrmdefault { \token_to_str:N \CJKrmdefault }
1182         \CJKsfdefault { \token_to_str:N \CJKsfdefault }
1183         \CJKttdefault { \token_to_str:N \CJKttdefault }
1184     }
1185     {#1}
1186 }

\ctex_ltj_fontspec:nn 1187 \cs_new_protected:Npn \ctex_ltj_fontspec:nn #1#2
1188 {
1189     \prop_get:NnNTF \g__ctex_ltj_fontspec_prop
1190     { CJKfontspec/#1/#2/id } \l_ctex_ltj_family_tl
1191     { \ctex_ltj_switch_family:x { \l_ctex_ltj_family_tl } }
1192     {
1193         \int_gincr:N \g__ctex_ltj_family_int
1194         \__ctex_ltj_fontspec:enn
1195         { CJKfontspec ( \int_use:N \g__ctex_ltj_family_int ) }
1196         {#1} {#2}
1197     }
1198 }
1199 \cs_new_protected:Npn \ctex_ltj_fontspec:ee #1#2
1200 { \use:e { \ctex_ltj_fontspec:nn {#1} {#2} } }
1201 \cs_new_protected:Npn \__ctex_ltj_fontspec:nnn #1#2#3
1202 {
1203     \bool_if:NT \l__ctex_ltj_add_alternate_bool
1204     {
1205         \cs_if_free:cF
1206         { \__ctex_ltj_alternate_cs:n { reset / \l_ctex_ltj_family_tl } }
1207         {
1208             \cs_gset_eq:cc
1209             { \__ctex_ltj_alternate_cs:n { reset / #1 } }
1210             { \__ctex_ltj_alternate_cs:n { reset / \l_ctex_ltj_family_tl } }
1211             \cs_gset_eq:cc
1212             { \__ctex_ltj_alternate_cs:n { clear / #1 } }
1213             { \__ctex_ltj_alternate_cs:n { clear / \l_ctex_ltj_family_tl } }
1214         }
1215         \bool_set_false:N \l__ctex_ltj_add_alternate_bool
1216     }
1217     \prop_gput:Nnn \g__ctex_ltj_fontspec_prop { CJKfontspec/#2/#3/id } {#1}
1218     \ctex_ltj_set_family:nnn {#1} {#2} {#3}
1219     \ctex_ltj_switch_family:n {#1}
1220 }
1221 \cs_generate_variant:Nn \__ctex_ltj_fontspec:nnn { e }
1222 \prop_new:N \g__ctex_ltj_fontspec_prop

\ctex_ltj_add_font_features:n 1223 \cs_new_protected:Npn \ctex_ltj_add_font_features:n #1
\ctex_ltj_add_font_features:nn 1224 { \ctex_ltj_add_font_features:xn { \l_ctex_ltj_family_tl } {#1} }
1225 \cs_new_protected:Npn \ctex_ltj_add_font_features:nn #1#2
1226 {
1227     \prop_get:NnNTF \g__ctex_ltj_family_font_name_prop
1228     {#1} \l__ctex_ltj_tmp_tl
1229     {
1230         \prop_get:NnN \g__ctex_ltj_family_font_options_prop
1231         {#1} \l__ctex_ltj_font_options_clist
1232         \clist_put_right:Nn \l__ctex_ltj_font_options_clist {#2}
1233         \bool_set_true:N \l__ctex_ltj_add_alternate_bool
1234         \ctex_ltj_fontspec:ee

```

```

1235         { \exp_not:o { \l__ctex_ltj_font_options_clist } }
1236         { \exp_not:o { \l__ctex_ltj_tmp_tl } }
1237     }
1238     { \msg_warning:nn { ctex } { addCJKfontfeature-ignored } }
1239 }
1240 \bool_new:N \l__ctex_ltj_add_alternate_bool
1241 \cs_generate_variant:Nn \ctex_ltj_add_font_features:n { x }
1242 \cs_generate_variant:Nn \ctex_ltj_add_font_features:nn { x }
1243 \msg_new:nnn { ctex } { addCJKfontfeature-ignored }
1244 {
1245     \token_to_str:N \addCJKfontfeature (s)~ignored.\
1246     It~cannot~be~used~with~a~font~that~wasn't~selected~by~ctex.
1247 }

```

为了支持字体属性可选项在前在后两种语法,给出两个辅助工具,自带展开功能。

```

1248 \cs_new_protected:Npn \__ctex_ltj_pass_args:nnnn #1#2#3#4
1249 {
1250     \tl_if_novalue:nTF {#2}
1251     { \__ctex_ltj_post_arg:w {#1} {#3} {#4} }
1252     {
1253         \use:e { #1 {#2} {#3} }
1254         #4
1255     }
1256 }
1257 \NewDocumentCommand \__ctex_ltj_post_arg:w { m m m O { } }
1258 {
1259     \use:e { #1 {#4} {#2} }
1260     #3
1261 }

```

```

\setCJKfamilyfont 1262 \NewDocumentCommand \setCJKfamilyfont { m o m }
\newCJKfontfamily 1263 {
\CJKfontspec 1264     \__ctex_ltj_pass_args:nnnn
\addCJKfontfeatures 1265     { \ctex_ltj_set_family:nnn {#1} } {#2} {#3}
1266     { }
1267 }
1268 \NewDocumentCommand \newCJKfontfamily { o m o m }
1269 {
1270     \tl_set:Nx \l__ctex_ltj_tmp_tl
1271     { \tl_if_novalue:nTF {#1} { \cs_to_str:N #2 } {#1} }
1272     \cs_new_protected:Npx #2
1273     { \ctex_ltj_switch_family:n { \l__ctex_ltj_tmp_tl } }
1274     \__ctex_ltj_pass_args:nnnn
1275     { \ctex_ltj_set_family:nnn { \l__ctex_ltj_tmp_tl } } {#3} {#4}
1276     { }
1277 }
1278 \NewDocumentCommand \CJKfontspec { o m }
1279 {
1280     \__ctex_ltj_pass_args:nnnn
1281     { \ctex_ltj_fontspeg:nn {#1} {#2}
1282     { \tex_ignorespaces:D }
1283     }
1284 \NewDocumentCommand \addCJKfontfeatures { m }
1285 {
1286     \ctex_ltj_add_font_features:x {#1}
1287     \tex_ignorespaces:D
1288 }
1289 \cs_new_eq:NN \addCJKfontfeature \addCJKfontfeatures

\setCJKmainfont 1290 \NewDocumentCommand \setCJKmainfont { o m }
\setCJKsansfont 1291 {
\setCJKmonofont 1292     \__ctex_ltj_pass_args:nnnn
\setCJKmathfont 1293     { \ctex_ltj_set_family:nnn { \CJKrmdefault } } {#1} {#2}
\defaultCJKfontfeatures 1294     { \normalfont }
1295 }

```

```

1296 \cs_new_eq:NN \setCJKromanfont \setCJKmainfont
1297 \NewDocumentCommand \setCJKsansfont { o m }
1298 {
1299   \__ctex_ltj_pass_args:nnnn
1300   { \ctex_ltj_set_family:nnn { \CJKsfdefault } } {#1} {#2}
1301   { \normalfont }
1302 }
1303 \NewDocumentCommand \setCJKmonofont { o m }
1304 {
1305   \__ctex_ltj_pass_args:nnnn
1306   { \ctex_ltj_set_family:nnn { \CJKttdefault } } {#1} {#2}
1307   { \normalfont }
1308 }
1309 \NewDocumentCommand \setCJKmathfont { o m }
1310 {
1311   \__ctex_ltj_pass_args:nnnn
1312   { \ctex_ltj_set_family:nnn { \c__ctex_ltj_math_tl } } {#1} {#2}
1313   { }
1314 }
1315 \NewDocumentCommand \defaultCJKfontfeatures { m }
1316 { \clist_gset:Nn \g__ctex_ltj_default_features_clist {#1} }
1317 \clist_new:N \g__ctex_ltj_default_features_clist
1318 \@onlypreamble \setCJKmainfont
1319 \@onlypreamble \setCJKsansfont
1320 \@onlypreamble \setCJKmonofont
1321 \@onlypreamble \setCJKmathfont
1322 \@onlypreamble \setCJKromanfont
1323 \@onlypreamble \defaultCJKfontfeatures

1324 \tl_if_exist:NF \CJKfamilydefault
1325 { \tl_const:Nn \CJKfamilydefault { \CJKrmdefault } }
1326 \tl_if_exist:NF \CJKrmdefault { \tl_const:Nn \CJKrmdefault { rm } }
1327 \tl_if_exist:NF \CJKsfdefault { \tl_const:Nn \CJKsfdefault { sf } }
1328 \tl_if_exist:NF \CJKttdefault { \tl_const:Nn \CJKttdefault { tt } }

```

使用 $\text{\LaTeX}_2\epsilon$ 提供的钩子,使主要 CJK 字体族能随西文主要字体更新。

```

1329 \g@addto@macro \@rmfamilyhook { \CJKfamily { \CJKrmdefault } }
1330 \g@addto@macro \@sffamilyhook { \CJKfamily { \CJKsfdefault } }
1331 \g@addto@macro \@ttfamilyhook { \CJKfamily { \CJKttdefault } }
1332 \g@addto@macro \@defaultfamilyhook { \CJKfamily { \CJKfamilydefault } }

```

在导言区末尾更新 \CJKfamilydefault。

```

1333 \ctex_at_end_preamble:n { \ctex_update_default_family: }

```

\ctex_ltj_ensure_default_family: 在导言区结束确认 \CJKfamilydefault 确实存在。

```

1334 \cs_new_protected:Npn \ctex_ltj_ensure_default_family:
1335 {
1336   \prop_if_empty:NF \g__ctex_ltj_family_font_name_prop
1337   {
1338     \ctex_ltj_family_if_exist:xNF { \CJKfamilydefault } \l__ctex_ltj_tmp_tl
1339     {
1340       \str_if_eq:eeTF { \CJKfamilydefault } { \CJKrmdefault }
1341       { \use:n }
1342       {
1343         \ctex_ltj_family_if_exist:xNTF { \CJKrmdefault } \l__ctex_ltj_tmp_tl
1344         { \tl_gset:Nn \CJKfamilydefault { \CJKrmdefault } \use_none:n }
1345         { \use:n }
1346       }
1347     }
1348     \prop_map_inline:Nn \g__ctex_ltj_family_font_name_prop
1349     {
1350       \prop_map_break:n
1351       { \tl_gset_rescan:Nnn \CJKfamilydefault { } { ##1 } }
1352     }
1353   }

```

```

1354     }
1355     \normalfont
1356     \ctex_ltj_update_mathfont:
1357   }
1358 }

```

\ctex_ltj_update_mathfont: 更新数学字体为实际的字体。

```

1359 \cs_new_protected:Npn \ctex_ltj_update_mathfont:
1360 {
1361   \ctex_ltj_family_if_exist:xNTF { \c__ctex_ltj_math_tl } \l__ctex_ltj_tmp_tl
1362   { \ctex_ltj_update_mathfont:n { \l__ctex_ltj_tmp_tl } }
1363   {
1364     \ctex_ltj_family_if_exist:xNT { \CJKfamilydefault } \l__ctex_ltj_tmp_tl
1365     { \ctex_ltj_update_mathfont:n { \l__ctex_ltj_tmp_tl } }
1366   }
1367 }
1368 \cs_new_protected:Npn \ctex_ltj_update_mathfont:n #1
1369 {
1370   \tl_const:Nx \c__ctex_ltj_math_family_tl {#1}
1371   \DeclareSymbolFont { \c__ctex_ltj_math_tl } { \CJK@encoding }
1372   { \c__ctex_ltj_math_family_tl } { \mddefault } { \shapedefault }
1373   \cs_if_free:cTF
1374   { \CJK@encoding/\c__ctex_ltj_math_family_tl/\bfdefault/\shapedefault }
1375   {
1376     \SetSymbolFont { \c__ctex_ltj_math_tl } { bold } { \CJK@encoding }
1377     { \c__ctex_ltj_math_family_tl } { \mddefault } { \shapedefault }
1378   }
1379   {
1380     \SetSymbolFont { \c__ctex_ltj_math_tl } { bold } { \CJK@encoding }
1381     { \c__ctex_ltj_math_family_tl } { \bfdefault } { \shapedefault }
1382   }
1383 }

```

14.3.4.7 替代字体的设置

AlternateFont
CharRange

设置替代字体的选项。

```

1384 \keys_define:nn { ctex_ltj / fontspec }
1385 {
1386   AlternateFont .code:n = \ctex_ltj_set_alternate_prop:n {#1} ,
1387   AlternateFont .value_required:n = true ,
1388   CharRange .clist_set:N = \l__ctex_ltj_char_range_clist ,
1389   CharRange .value_required:n = true
1390 }

```

\ctex_ltj_set_alternate_prop:n 保存替代字体序列。

```

1391 \cs_new_protected:Npn \ctex_ltj_set_alternate_prop:n #1
1392 { \clist_map_function:nN {#1} \__ctex_ltj_push_alternate_prop:n }
1393 \cs_new_protected:Npn \__ctex_ltj_push_alternate_prop:n #1
1394 {
1395   \clist_set:Nx \l__ctex_ltj_tmp_clist { \tl_head:n {#1} }
1396   \tl_remove_all:Nn \l__ctex_ltj_tmp_clist { ~ }
1397   \exp_args:No \__ctex_ltj_push_alternate_prop:nn
1398   { \l__ctex_ltj_tmp_clist } {#1}
1399 }
1400 \cs_new_protected:Npn \__ctex_ltj_push_alternate_prop:nn #1
1401 {
1402   \prop_remove:Nn \l__ctex_ltj_alternate_prop {#1}
1403   \prop_put:Nnn \l__ctex_ltj_alternate_prop {#1}
1404 }
1405 \clist_new:N \l__ctex_ltj_tmp_clist
1406 \prop_new:N \l__ctex_ltj_alternate_prop

```

\ctex_ltj_set_altername_family:nnF

如果在字体的选项中设置了 CharRange, 则只设置替代字体。

```

1407 \cs_new_protected:Npn \ctex_ltj_set_altername_family:nnF
1408 {
1409   \clist_if_empty:NTF \l__ctex_ltj_char_range_clist
1410   { \__ctex_ltj_set_family_aux:nnn }
1411   { \__ctex_ltj_set_altername_family_aux:nnn }
1412 }
1413 \cs_new_protected:Npn \__ctex_ltj_set_family_aux:nnn #1#2#3
1414 {
1415   \__ctex_ltj_check_family:n {#1}
1416   \prop_if_empty:NF \l__ctex_ltj_altername_prop
1417   { \ctex_ltj_save_altername_seq:cn { \__ctex_ltj_altername_cs:n {#1} } {#2} }
1418   #3
1419 }
1420 \cs_new_protected:Npn \__ctex_ltj_set_altername_family_aux:nnn #1#2#3
1421 { \ctex_ltj_set_altername_family:nn {#1} {#2} }

```

\ctex_ltj_save_altername_seq:Nn

保存由 AlternateFont 设置的替代字体序列。

```

1422 \cs_new_protected:Npn \ctex_ltj_save_altername_seq:Nn #1#2
1423 {
1424   \prop_map_inline:Nn \l__ctex_ltj_altername_prop
1425   { \__ctex_ltj_save_altername_auxi:w ##2 { } \q_mark #1 {#2} }
1426 }
1427 \cs_new_protected:Npn \__ctex_ltj_save_altername_auxi:w #1#2#
1428 {
1429   \tl_if_blank:nTF {#2}
1430   { \__ctex_ltj_save_altername_auxii:w {#1} }
1431   { \__ctex_ltj_save_altername_auxii:w {#1} {#2} }
1432 }
1433 \cs_new_protected:Npn \__ctex_ltj_save_altername_auxii:w #1#2#3 #4 \q_mark #5#6
1434 {
1435   \clist_set:Nn \l__ctex_ltj_char_range_clist {#1}
1436   \clist_set:Nn \l__ctex_ltj_altername_options_clist {#3}
1437   \__ctex_ltj_use_global_options:N \l__ctex_ltj_altername_options_clist
1438   \tl_if_blank:nTF {#2}
1439   { \tl_set:Nn \l__ctex_ltj_tmp_tl {#6} }
1440   {
1441     \tl_set:Nx \l__ctex_ltj_tmp_tl { \tl_trim_spaces:n {#2} }
1442     \tl_replace_all:Nnn \l__ctex_ltj_tmp_tl { * } {#6}
1443   }
1444   \use:e
1445   {
1446     \ctex_ltj_save_altername_family:Nnnn \exp_not:N #5
1447     { \exp_not:o { \l__ctex_ltj_char_range_clist } }
1448     { \exp_not:o { \l__ctex_ltj_altername_options_clist } }
1449     { \exp_not:o { \l__ctex_ltj_tmp_tl } }
1450   }
1451 }
1452 \clist_new:N \l__ctex_ltj_altername_options_clist
1453 \cs_generate_variant:Nn \ctex_ltj_save_altername_seq:Nn { c }

```

\ctex_ltj_set_altername_family:nn

设置选项 CharRange 范围内的替代字体。如果已经定义了主字体, 我们也马上定义替代字体, 否则只保存起来备用。

```

1454 \cs_new_protected:Npn \ctex_ltj_set_altername_family:nn #1#2
1455 {
1456   \__ctex_ltj_update_family_uid:N \l__ctex_ltj_font_options_clist
1457   \__ctex_ltj_use_global_options:N \l__ctex_ltj_font_options_clist
1458   \ctex_ltj_set_altername_family:coonn
1459   { \__ctex_ltj_altername_cs:n {#1} }
1460   { \l__ctex_ltj_char_range_clist }
1461   { \l__ctex_ltj_font_options_clist } {#2} {#1}
1462 }
1463 \cs_new_protected:Npn \ctex_ltj_set_altername_family:Nnnnn #1#2#3#4#5
1464 {

```

```

1465 \prop_get:NnNT \g__ctex_ltj_family_name_prop {#5} \l__ctex_ltj_base_family_tl
1466 { \ctex_ltj_set_alternate_family:nnn {#2} {#3} {#4} }
1467 \ctex_ltj_save_alternate_family:Nnnn #1 {#2} {#3} {#4}
1468 }
1469 \cs_generate_variant:Nn \ctex_ltj_set_alternate_family:Nnnnn { coo }

```

保存替代字体序列的定义,以备定义主字体时使用。

```

1470 \cs_new_protected:Npn \ctex_ltj_save_alternate_family:Nnnn #1#2#3#4
1471 {
1472   \cs_if_exist:NF #1 { \cs_set_eq:NN #1 \prg_do_nothing: }
1473   \cs_gset_protected:Npx #1
1474     { \exp_not:o { #1 \ctex_ltj_set_alternate_family:nnn {#2} {#3} {#4} } }
1475 }

```

实际定义替代字体族。

```

1476 \cs_new_protected:Npn \ctex_ltj_set_alternate_family:nnn #1#2#3
1477 {
1478   \group_begin:
1479   \cs_set_eq:NN \CJKfamily \use_none:n
1480   \ctex_ltj_swap_cs:NN
1481   \DeclareFontShape@ \ctex_ltj_declare_alternate_shape:nnnnnn
1482   \tl_set:Nn \l__ctex_ltj_char_range_clist {#1}
1483   \fontspec_set_family:Nnn \l__ctex_ltj_alternate_family_tl {#2} {#3}
1484   \group_end:
1485 }
1486 \tl_new:N \l__ctex_ltj_alternate_family_tl

```

交换两个控制序列的意义。

```

1487 \cs_new_protected:Npn \ctex_ltj_swap_cs:NN #1#2
1488 {
1489   \cs_set_eq:NN \__ctex_ltj_tmp:w #1
1490   \cs_set_eq:NN #1 #2
1491   \cs_set_eq:NN #2 \__ctex_ltj_tmp:w
1492   \cs_undefine:N \__ctex_ltj_tmp:w
1493 }

```

fontspec 在一个字体族的选项和字体名称相同的时候,就不定义新字体。为了避免混淆替代字体的设置,我们新定义一个虚拟的选项 **LTJFONTUID**,确保 **fontspec** 对 **CJK** 字体族总是定义新字体。

```

1494 \keys_define:nn { fontspec } { LTJFONTUID .code:n = }
1495 \cs_new_protected:Npn \__ctex_ltj_update_family_uid:N #1
1496 {
1497   \int_gincr:N \g__ctex_ltj_family_int
1498   \clist_put_right:Nx #1 { LTJFONTUID = \int_use:N \g__ctex_ltj_family_int }
1499 }
1500 \int_new:N \g__ctex_ltj_family_int

```

在定义替代字体的字形时,通过字符范围与主字体的对应字形关联起来。**\DeclareFontShape@** 一个有六个参数,我们只需要使用它的第三个参数 *<series>* 和第四个参数 *<shape>*。

```

1501 \cs_new_protected:Npn \ctex_ltj_declare_alternate_shape:nnnnnn #1#2#3#4#5#6
1502 {
1503   \ctex_ltj_declare_alternate_shape:nnnnnn {#1} {#2} {#3} {#4} {#5} {#6}
1504   \ctex_ltj_set_alternate_shape:Nnnnnnn \l__ctex_ltj_char_range_clist
1505     { \l__ctex_ltj_base_family_tl } {#3} {#4} {#2} {#3} {#4}
1506 }

```

与 **LuaTeX-ja** 的 **\DeclareAlternateKanjiFont** 的功能类似,区别是固定编码为 **\CJK@encoding**。这个设置总是全局的。

```

1507 \cs_new_protected:Npn \ctex_ltj_set_alternate_shape:Nnnnnnn #1#2#3#4#5#6#7

```

```

1508 {
1509   \clist_map_inline:Nn #1
1510   {
1511     \prop_get:NnNTF \g__ctex_ltj_char_range_prop { ##1 } \l__ctex_ltj_char_range_tl
1512     {
1513       \ctex_ltj_set_alter_nate_shape:nnN { #2/#3/#4 } { #5/#6/#7 }
1514       \l__ctex_ltj_char_range_tl
1515     }
1516     { \ctex_ltj_set_alter_nate_shape:nnn { #2/#3/#4 } { #5/#6/#7 } { ##1 } }
1517   }
1518   \__ctex_ltj_save_alter_nate_shape:cn
1519   { \__ctex_ltj_alter_nate_cs:n { clear / \l__ctex_ltj_base_CJKfamily_tl } }
1520   { \ctex_ltj_clear_alt_font:n { \CJK@encoding/#2/#3/#4 } }
1521 }

```

\ctex_ltj_set_alter_nate_shape:nnn

我们使用 `->` 而不是像 `LuaTeX-j`a 一样使用 `-` 作为区间的分隔符。`LuaTeX-j`a 支持使用负数来引用由 `JFM` 设置的字符类。如果使用 `-` 作为分隔符,那么负数单独使用时,就需要把它放在两层花括号之内(例如 `{-1}`}),或者使用类似 `{-1}{-1}` 的形式才不会解释错误。

```

1522 \NewDocumentCommand \ctex_ltj_set_alter_nate_shape:nnn
1523 { m m > { \SplitArgument { 1 } { -> } } m }
1524 { \ctex_ltj_set_alter_nate_shape:nnnn {#1} {#2} #3 }
1525 \cs_new_protected:Npn \ctex_ltj_set_alter_nate_shape:nnnn #1#2#3#4
1526 {
1527   \ctex_ltj_set_alter_nate_shape:e
1528   {
1529     \__ctex_ltj_range_normalization:nn {#3} {#4}
1530     { \CJK@encoding / \exp_not:n {#2} }
1531     { \CJK@encoding / \exp_not:n {#1} }
1532   }
1533 }
1534 \cs_new_protected:Npn \ctex_ltj_set_alter_nate_shape:n #1
1535 {
1536   \ctex_ltj_set_alt_font:nnnn #1
1537   \__ctex_ltj_save_alter_nate_shape:cn
1538   { \__ctex_ltj_alter_nate_cs:n { reset / \l__ctex_ltj_base_CJKfamily_tl } }
1539   { \ctex_ltj_set_alt_font:nnnn #1 }
1540 }
1541 \cs_generate_variant:Nn \ctex_ltj_set_alter_nate_shape:n { e }

```

\ctex_ltj_set_alter_nate_shape:nnN

若字符范围预先由 `declarecharrange` 声明,则可以直接使用。

```

1542 \cs_new_protected:Npn \ctex_ltj_set_alter_nate_shape:nnN #1#2#3
1543 {
1544   \tl_map_inline:Nn #3
1545   {
1546     \ctex_ltj_set_alter_nate_shape:n
1547     {
1548       ##1
1549       { \CJK@encoding/#2 }
1550       { \CJK@encoding/#1 }
1551     }
1552   }
1553 }

```

__ctex_ltj_save_alter_nate_shape:Nn

将实际设置的替换字形保存起来用于清除或恢复。暂时令 `\l__ctex_ltj_base_family_tl` 为 `\scan_stop:` 是让它不被展开,使得替换字体的设置可以在 `\addCJKfontfeature` 中直接使用。

```

1554 \cs_new_protected:Npn \__ctex_ltj_save_alter_nate_shape:Nn #1#2
1555 {
1556   \group_begin:
1557   \cs_if_exist:NF #1 { \cs_set_eq:NN #1 \prg_do_nothing: }
1558   \cs_set_eq:NN \l__ctex_ltj_base_family_tl \scan_stop:
1559   \cs_gset_protected:Npx #1 { \exp_not:o {#1} #2 }

```

```

1560 \group_end:
1561 }
1562 \cs_generate_variant:Nn \__ctex_ltj_save_alter_nate_shape:Nn { c }

```

clearalter_natefont
resetalter_natefont

清除和重置操作总是全局的。

```

1563 \ctex_define:n
1564 {
1565   clearalter_natefont .code:n =
1566   { \clist_map_function:eN {#1} \ctex_ltj_clear_alter_nate_font:n } ,
1567   resetalter_natefont .code:n =
1568   { \clist_map_function:eN {#1} \ctex_ltj_reset_alter_nate_font:n } ,
1569   clearalter_natefont .default:n = \l_ctex_ltj_family_tl ,
1570   resetalter_natefont .default:n = \l_ctex_ltj_family_tl
1571 }
1572 \cs_new_protected:Npn \ctex_ltj_clear_alter_nate_font:n #1
1573 {
1574   \group_begin:
1575   \ctex_ltj_family_if_exist:xNTF {#1} \l__ctex_ltj_base_family_tl
1576   {
1577     \cs_if_exist_use:cT
1578     { \__ctex_ltj_alter_nate_cs:n { clear / #1 } }
1579     {
1580       \prop_gput:Nno \g__ctex_ltj_reset_alter_nate_prop
1581       {#1} { \l__ctex_ltj_base_family_tl }
1582       \tl_set_eq:NN \CJK@family \l__ctex_ltj_base_family_tl
1583       \selectfont
1584     }
1585   }
1586   { \__ctex_ltj_family_unknown_warning:n {#1} }
1587   \group_end:
1588 }
1589 \cs_new_protected:Npn \ctex_ltj_reset_alter_nate_font:n #1
1590 {
1591   \group_begin:
1592   \prop_gpop:NnNT \g__ctex_ltj_reset_alter_nate_prop {#1} \CJK@family
1593   {
1594     \tl_set_eq:NN \l__ctex_ltj_base_family_tl \CJK@family
1595     \use:c { \__ctex_ltj_alter_nate_cs:n { reset / #1 } }
1596     \selectfont
1597   }
1598   \group_end:
1599 }
1600 \prop_new:N \g__ctex_ltj_reset_alter_nate_prop
1601 \cs_generate_variant:Nn \clist_map_function:nN { e }

```

declarecharrange

预先声明字符范围。

```

1602 \ctex_define:n
1603 {
1604   declarecharrange .code:n = \ctex_ltj_declare_char_range:e {#1} ,
1605   declarecharrange .value_required:n = true
1606 }
1607 \cs_new_protected:Npn \ctex_ltj_declare_char_range:n #1
1608 { \clist_map_inline:nn {#1} { \__ctex_ltj_declare_char_range:nn ##1 } }
1609 \cs_generate_variant:Nn \ctex_ltj_declare_char_range:n { e }
1610 \cs_new_protected:Npn \__ctex_ltj_declare_char_range:nn #1
1611 { \exp_args:Ne \ctex_ltj_declare_char_range:nn { \tl_trim_spaces:n {#1} } }

```

#1 是名字, #2 是范围。

```

1612 \cs_new_protected:Npn \ctex_ltj_declare_char_range:nn #1#2
1613 {
1614   \tl_clear:N \l__ctex_ltj_char_range_tl
1615   \clist_map_function:nN {#2} \ctex_ltj_save_char_range:n
1616   \prop_gput:Nno \g__ctex_ltj_char_range_prop {#1} { \l__ctex_ltj_char_range_tl }
1617   \ctex_ltj_def_char_range_key:n {#1}

```

\ctex_ltj_declare_char_range:nn
\g__ctex_ltj_char_range_prop


```

1618 \tl_clear:N \l__ctex_ltj_char_range_tl
1619 }
1620 \tl_new:N \l__ctex_ltj_char_range_tl
1621 \prop_new:N \g__ctex_ltj_char_range_prop

```

\ctex_ltj_save_char_range:n 预先解释字符区间的意义。

```

1622 \NewDocumentCommand \ctex_ltj_save_char_range:n
1623 { > { \SplitArgument { 1 } { -> } } m }
1624 { \ctex_ltj_save_char_range:nn #1 }
1625 \cs_new_protected:Npn \ctex_ltj_save_char_range:nn #1#2
1626 {
1627   \tl_put_right:Nx \l__ctex_ltj_char_range_tl
1628     { { \__ctex_ltj_range_normalization:nn {#1} {#2} } }
1629 }
1630 \cs_new:Npn \__ctex_ltj_range_normalization:nn #1#2
1631 {
1632   \tl_if_novalue:nTF {#2}
1633   {
1634     { \int_eval:n {#1} }
1635     { \int_eval:n {#1} }
1636   }
1637   {
1638     { \int_eval:n { \tl_if_blank:nTF {#1} { \c__ctex_ltj_range_min_int } {#1} } }
1639     { \int_eval:n { \tl_if_blank:nTF {#2} { \c__ctex_ltj_range_max_int } {#2} } }
1640   }
1641 }
1642 \int_const:Nn \c__ctex_ltj_range_min_int { "80 }
1643 \int_const:Nn \c__ctex_ltj_range_max_int { \c_max_char_int }

```

\ctex_ltj_def_char_range_key:n 在字体设置选项中定义字符范围键。

```

1644 \cs_new_protected:Npn \ctex_ltj_def_char_range_key:n #1
1645 {
1646   \keys_if_exist:nnF { ctex_ltj / fontspec } {#1}
1647   {
1648     \keys_define:nn { ctex_ltj / fontspec }
1649     { #1 .code:n = \ctex_ltj_char_range_key:nn {#1} { ##1 } }
1650   }
1651 }

```

\ctex_ltj_char_range_key:nn 如果字符范围键没有值,则只设置的这个字符范围内的替代字体。

```

1652 \cs_new_protected:Npn \ctex_ltj_char_range_key:nn #1#2
1653 {
1654   \tl_if_blank:nTF {#2}
1655   { \clist_set:Nn \l__ctex_ltj_char_range_clist {#1} }
1656   { \__ctex_ltj_push_alternate_prop:nn {#1} { {#1} #2 } }
1657 }

```

14.3.4.8 其他设置

在抄录环境中禁用 autospacing 和 autoxspacing。然而, LuaTeX-jā 还是会使 JAchar 自动折行。没有看到有简单的禁用折行的办法,可能需要设置所有的 JAchar 的 prebreakpenalty 或 postbreakpenalty 为 10000:

```

\directlua
{
  luatexja.isglobal = tex.globaldefs > 0 and "global" or ""
  for i = 0x80, 0x10FFFF do
    if luatexja.charrange.jcr_table_main[i] > 0 and
      luatexja.charrange.jcr_table_main[i] < 218 and
      luatexja.charrange.is_japanese_char_curlist(i) then
      luatexja.stack.set_stack_table(luatexja.stack_table_index.PRE + i, 10000)
    end
  end
}

```

```

        end
    }

1658 \AtBeginDocument
1659 {
1660     \ctex_appto_cmd:NnnTF \verbatim@font
1661     { \char_set_catcode_letter:n { 64 } }
1662     { \CTEX@verbatim@font@hook }
1663     { }
1664     { \ctex_patch_failure:N \verbatim@font }
1665 }
1666 \cs_new_protected:Npn \CTEX@verbatim@font@hook
1667 { \ltjsetparameter { autospacing = false , autoxspacing = false } }

@@@italiccorr LATEX 的倾斜校正也要重新定义。
1668 \cs_set_eq:NN \@@italiccorr \/

```

`\ctex_ltj_set_kanjiskip:N` 和 `\ctex_ltj_set_xkanjiskip:N` 是相应的 `\ltjsetparameter` 的快捷方式, 在使用他们时, 要注意先使用 `\ltjsetpar@global`。

```

1669 \cs_new_protected:Npn \ctex_ltj_set_kanjiskip:N
1670 { \ltjsetpar@global \ltjsetkanjiskip }
1671 \cs_new_protected:Npn \ctex_ltj_set_xkanjiskip:N
1672 { \ltjsetpar@global \ltjsetxkanjiskip }

1673 <@@=ctex>
1674 </luatex>

```

14.3.5 ctex-engine-uptex.def

```
1675 <*uptex|aptex>
```

按 CJK 的命名习惯模拟部分命令, 并设置默认字体。

```

1676 \tl_if_exist:NF \CJKfamilydefault
1677 { \tl_const:Nn \CJKfamilydefault { \CJKrmdefault } }
1678 \tl_if_exist:NF \CJKrmdefault { \tl_const:Nn \CJKrmdefault { zhrm } }
1679 \tl_if_exist:NF \CJKsfdefault { \tl_const:Nn \CJKsfdefault { zhsf } }
1680 \tl_if_exist:NF \CJKttdefault { \tl_const:Nn \CJKttdefault { zhht } }
1681 \tl_set:Nn \kanjifamilydefault { \CJKfamilydefault }
1682 \g@addto@macro \rmfamilyhook { \kanjifamily { \CJKrmdefault } }
1683 \g@addto@macro \sffamilyhook { \kanjifamily { \CJKsfdefault } }
1684 \g@addto@macro \ttfamilyhook { \kanjifamily { \CJKttdefault } }
1685 \NewDocumentCommand \CJKfamily { m }
1686 { \kanjifamily {#1} \selectfont }

```

将 up^LA_TE_X 的默认字体由 mc 改为 zhrm, 并启用 \jfam。

```

1687 \DeclareErrorKanjifont {JY2}{zhrm}{m}{n}{10}
1688 \DeclareKanjisubstitution {JY2}{zhrm}{m}{n}
1689 \DeclareKanjisubstitution {JT2}{zhrm}{m}{n}
1690 \DeclareSymbolFont{mincho}{JY2}{zhrm}{m}{n}
1691 \SetSymbolFont{mincho}{bold}{JY2}{zhrm}{bx}{n}
1692 \jfam \symmincho

```

在导言区末尾更新 \CJKfamilydefault。

```
1693 \ctex_at_end_preamble:n { \ctex_update_default_family: }
```

使修改立刻生效, 保证导言区字体族正确。

```
1694 \normalfont
```

`\em` 取消 up^LA_TE_X 对 `\em` 使用 `\mcfamily`、`\gtfamily` 命令的重定义, 恢复 L^AT_EX 2_ε 对 `\em` 的原始定义。如果用户已经重定义了 `\em`, 则新定义保持不变。up^LA_TE_X 2016/05/07u00 的定义有所变化, 这一行为可以由用户通过 `platexrelease` 包改变, 需要分支处理。

```
1695 \ctex_patch_cmd_once:NnnTF \em
```

```

1696 { \ExplSyntaxOff }
1697 { \eminnershape \else \gtfamily \itshape }
1698 { \eminnershape \else \itshape }
1699 { }
1700 {
1701   \ctex_patch_cmd:Nnn \em
1702     { \mcfamily \upshape \else \gtfamily \itshape }
1703     { \eminnershape \else \itshape }
1704 }
1705 \cs_set_nopar:Npn \eminnershape { \upshape }

```

`\ctex_set_upfamily:nnn`

将 NFSS 字体族 #1 设置为 JFM 字体名 #2, 粗体形式字体名 #3。其中字体名形如 upzhserif, 不包括表示方向的后缀 -h 与 -v。粗体字体名为空时不设置该字形。本命令不设置字体映射, 需要复用已有的字体映射或另行设置。

```

1706 \cs_new_protected:Npn \ctex_set_upfamily:nnn #1#2#3
1707 {
1708   \DeclareKanjiFamily{JY2}{#1}{ }
1709   \DeclareKanjiFamily{JT2}{#1}{ }
1710   \DeclareFontShape{JY2}{#1}{m}{n}{<->~ #2-h}{ }
1711   \DeclareFontShape{JT2}{#1}{m}{n}{<->~ #2-v}{ }
1712   \tl_if_empty:nF {#3}
1713   {
1714     \DeclareFontShape{JY2}{#1}{b}{n}{<->~ #3-h}{ }
1715     \DeclareFontShape{JT2}{#1}{b}{n}{<->~ #3-v}{ }
1716     \DeclareFontShape{JY2}{#1}{bx}{n}{<->~ #3-h}{ }
1717     \DeclareFontShape{JT2}{#1}{bx}{n}{<->~ #3-v}{ }
1718   }
1719 }

```

`\ctex_set_upmap:nnn`

设置 upTeX 字体映射。#1 是形如 upserif 的 PS TFM 字体名, 不带表示粗体的后缀 b 与表示排版方向的后缀 -h 与 -v。#2 与 #3 是普通与粗体的实际字体名。

```

1720 \cs_new_protected:Npn \ctex_set_upmap:nnn #1#2#3
1721 {
1722   \ctex_set_zhmap:n
1723   {
1724     \special { pdf:mapline~#1-h~UniGB-UTF16-H~#2 }
1725     \special { pdf:mapline~#1-v~UniGB-UTF16-V~#2 }
1726     \tl_if_empty:nF {#3}
1727     {
1728       \special { pdf:mapline~#1b-h~UniGB-UTF16-H~#3 }
1729       \special { pdf:mapline~#1b-v~UniGB-UTF16-V~#3 }
1730     }
1731   }
1732 }

```

`\ctex_set_upmap_unicode:nnn`

设置 upTeX 字体映射, 使用 unicode CMap。参数同上。

```

1733 \cs_new_protected:Npn \ctex_set_upmap_unicode:nnn #1#2#3
1734 {
1735   \ctex_set_zhmap:n
1736   {
1737     \special { pdf:mapline~#1-h~unicode~#2 }
1738     \special { pdf:mapline~#1-v~unicode~#2 }
1739     \tl_if_empty:nF {#3}
1740     {
1741       \special { pdf:mapline~#1b-h~unicode~#3 }
1742       \special { pdf:mapline~#1b-v~unicode~#3 }
1743     }
1744   }
1745 }

```

`\ctex_set_upfonts:nnnnnn`

设置 upTeX 基本字体映射, 按 zhmetrics-uptex 的定义, 依次设置衬线体正、粗、意大利, 无衬

线体正、粗，等宽体正——共 6 种字体，并分横排及直排。

```

1746 \cs_new_protected:Npn \ctex_set_upfonts:nnnnnn #1#2#3#4#5#6
1747 {
1748   \ctex_set_upmap:nnn { upserif } {#1} {#2}
1749   \ctex_set_upmap:nnn { upserifit } {#3} {}
1750   \ctex_set_upmap:nnn { upsans } {#4} {#5}
1751   \ctex_set_upmap:nnn { upmono } {#6} {}
1752 }

```

以下命令只能在导言区使用。

```

1753 \@onlypreamble \ctex_set_upfamily:nnn
1754 \@onlypreamble \ctex_set_upmap:nnn
1755 \@onlypreamble \ctex_set_upmap_unicode:nnn
1756 \@onlypreamble \ctex_set_upfonts:nnnnnn

```

`everyysel` 宏包 (2011/10/28) 未考虑 `upLATEX` 对 `\selectfont` 的修改，需要引入 `pxeveryysel` 宏包。

```

1757 \RequirePackage { pxeveryysel }

1758 </uptex|aptex>

```

14.3.6 调整 `\CJKfamilydefault`

`\ctex_update_default_family:` 在导言区结束，如果 `\CJKfamilydefault` 没有被更改，则在此时根据西文字体的情况更新 `\CJKfamilydefault`。`xeCJK` 已经有这个功能，不需要再调整。

```

1759 <*pdf|luatex|uptex|aptex>
1760 \cs_new_protected:Npn \ctex_update_default_family:
1761 {
1762   \tl_if_eq:NNT \CJKfamilydefault \l__ctex_family_default_init_tl
1763   {
1764     \group_begin:
1765     \cs_set_eq:NN \__ctex_family_default_wrap:n \exp_not:n
1766     \tl_gset:Nx \CJKfamilydefault
1767     {
1768       \str_case:onF { \familydefault }
1769       {
1770         { \rmdefault } { \exp_not:N \CJKrmdefault }
1771         { \sfdefault } { \exp_not:N \CJKsfdefault }
1772         { \ttdefault } { \exp_not:N \CJKttdefault }
1773       }
1774       { \CJKfamilydefault }
1775     }
1776     \group_end:
1777   }

```

使用 `LuaLATEX` 时，自动调整得到的 `\CJKfamilydefault` 可能没有定义，需要确认它的存在性。使用 `CJK` 宏包或 `upLATEX` 时，`C19rm`、`JY2rm` 等总是有定义的，不需要确认。

```

1778 <*luatex>
1779   \ctex_ljtj_ensure_default_family:
1780 </luatex>
1781 }

```

`\l__ctex_family_default_init_tl` 往 `\CJKfamilydefault` 中加入标志，用于判断它是否被更改。

```

1782 \tl_new:N \l__ctex_family_default_init_tl
1783 \cs_new_eq:NN \__ctex_family_default_wrap:n \use:n
1784 \tl_set:Nx \l__ctex_family_default_init_tl
1785 {
1786   \exp_not:N \__ctex_family_default_wrap:n
1787   { \exp_not:o { \CJKfamilydefault } }
1788 }

```

```

1789 \tl_gset_eq:NN \CJKfamilydefault \l__ctex_family_default_init_tl
1790 </pdfTeX|luatex|uptex|aptex>

```

14.3.7 hyperref 兼容性处理

现在处理各个引擎下的 PDF 中文书签问题。根据编译引擎与文件编码的不同, `ctex` 向 `hyperref` 传递适当的参数, 完成中文书签的正确设置。用户仍需要自己载入 `hyperref` 宏包。

`\ctex_hypersetup:n` 如果已经载入 `hyperref` 宏包, 则直接使用其定义设置选项; 否则 `\ctex_hypersetup:n` 的效果与 `\PassOptionsToPackage` 一致, 只传递宏包参数。如果用户不载入 `hyperref` 宏包, 相关参数即被丢弃。

```

1791 <*class|ctex>
1792 \ifpackageloaded { hyperref }
1793 {
1794   \cs_new_protected:Npn \ctex_hypersetup:n #1
1795     { \hypersetup {#1} }
1796 }
1797 {
1798   \cs_new_protected:Npn \ctex_hypersetup:n #1
1799     { \PassOptionsToPackage {#1} { hyperref } }
1800 }
1801 </class|ctex>

```

在 `pdfTeX` 下使用 GBK 编码, `DVIPDFMx` 驱动可以直接用它的 `\special` 命令, 其他模式用 `xCJK2uni` 宏包处理。使用 UTF-8 编码时, `CJKutf8` 已经处理了书签问题, 但仍需要设置 `pdfencoding` 为 `unicode`, 目的是在书签的开头写入 BOM (`\376\377`), 提示这是 UTF-16BE 字节流。

```

1802 <*pdfTeX>
1803 \ctex_hypersetup:n { driverfallback = dvipdfmx }
1804 \str_if_eq:onTF { \g__ctex_encoding_tl } { GBK }
1805 {
1806   \ctex_hypersetup:n { CJKbookmarks = true }
1807   \sys_if_output_pdf:TF
1808     { \ctex_at_end_package:nn { hyperref } { \RequirePackage { xCJK2uni } } }
1809     {
1810       \ctex_at_end_package:nn { hyperref }
1811       {
1812         \str_if_eq:onTF { \Hy@driver } { hdvipdfm }
1813         {
1814           \AtBeginShipoutFirst
1815             { \special { pdf:tounicode~GBK-EUC-UCS2 } }
1816         }
1817         { \RequirePackage { xCJK2uni } }
1818       }
1819     }
1820 }
1821 { \ctex_hypersetup:n { pdfencoding = unicode } }
1822 </pdfTeX>

```

在 `XYTeX` 下, `hyperref` 在处理带有非 ASCII 字符和 `\%` 的书签时有问题¹⁹。事实上, `hyperref` 在驱动文件 `hxdetex.def` 中设置了 `\Hy@unicodetrue`, 从而书签总是会被 `\HyPsd@ConvertToUnicode` 转化成 UTF-16BE 编码的形式(抄录自 `\pdfstringdef` 的定义):

```

\ifHy@unicode
  \HyPsd@ConvertToUnicode#1%
  \ifx\HyPsd@pdfencoding\HyPsd@pdfencoding@auto
    \ltx@ifundefined{StringEncodingConvertTest}{%
    }{%

```

¹⁹<https://github.com/CTeX-org/ctex-kit/issues/39>

```

\edef\unescapestring\HyPsd@temp#1%
\ifxetex
\let\HyPsd@UnescapedString\HyPsd@temp
\StringEncodingConvertTest\HyPsd@temp\HyPsd@temp
{utf16be}{ascii-print}{%
\edef\unescapestring\HyPsd@temp\HyPsd@temp
\global\let#1\HyPsd@temp
\HyPsd@EscapeTeX#1%
\Hy@unicodetrue
}{%
\HyPsd@ToBigChars#1%
}%

```

通过宏包选项 `pdfencoding=unicode` 设置 `\HyPsd@pdfencoding` 为 `unicode`, 可以避免随后再将书签从 UTF-16BE 字节流转化回正常字符 (其中使用的 `\HyPsd@ToBigChars` 没有考虑书签中含有 `\%` 的情况)。Heiko Oberdiek 在 README 中说明了将书签转化回正常字符的意图: 避免 XDVIPDFMX 的警告²⁰:

```
** WARNING ** Failed to convert input string to UTF16...
```

X_YTeX 的维护者 Khaled Hosny 已经注意到了这个问题²¹。需要注意的是, `hxeTeX.def` 重载了宏包选项 `unicode`, 目的是不能设置它为 `false`, 但也导致它不会改变 `\HyPsd@pdfencoding`。如果 `hyperref` 先于 CT_YTeX 被载入, 那么 `unicode` 选项是没有意义的。因此要通过意义相同但在 X_YTeX 下更保险的 `pdfencoding` 选项来设置。为了与 X_YTeX 下的行为一致 (使用 `\HyPsd@LoadUnicode` 载入 `puenc.def`), 在 LuaTeX 下也启用这个选项。

```

1823 <*xetex|luatex>
1824 \ctex_hypersetup:n { pdfencoding = unicode }
1825 </xetex|luatex>

```

我们假定 upTeX 使用 DVIPDFMX 驱动输出, 于是使用与 pdfTeX 类似的设置。注意 upTeX 需要使用 UTF8-UTF16 的编码转换。

```

1826 <*uptex|aptex>
1827 \ctex_hypersetup:n { driverfallback = dvipdfmx }
1828 \ctex_at_end_package:nn { hyperref }
1829 { \AtBeginShipoutFirst { \special { pdf:tounicode~UTF8-UTF16 } } }
1830 </uptex|aptex>
1831 <*pdfTeX|xetex|luatex|uptex|aptex>

```

14.3.8 CJKfntef、xeCJKfntef 相关设置

对 pdfTeX 与 X_YTeX 引擎, 分别在 CJKfntef、xeCJKfntef 宏包的末尾关闭彩色显式等多余格式。

```

1832 <*pdfTeX>
1833 \ctex_at_end_package:nn { CJKfntef }
1834 {
1835   \normalem
1836   \cs_new_protected:Npn \__ctex_clear_fntef_color:n #1
1837     { \tl_clear:c { CJK#1color } }
1838 </pdfTeX>
1839 <*xetex>
1840 \ctex_at_end_package:nn { xeCJKfntef }
1841 {
1842   \@ifpackagelater { xeCJKfntef } { 2014/11/04 }
1843   {
1844     \cs_new_protected:Npn \__ctex_clear_fntef_color:n #1
1845       { \xeCJKsetup { #1 / format = { } } }
1846   }

```

²⁰<http://project.ktug.org/dvipdfmx/mailman/dvipdfmx/2009-December/000153.html>

²¹<http://tug.org/pipermail/tex-live/2013-December/034613.html>

```

1847     {
1848         \cs_new_protected:Npn \__ctex_clear_fntef_color:n #1
1849         { \tl_clear:c { CJK#1color } }
1850     }
1851 </xetex>
1852 <*pdfTeX|xetex>
1853     \clist_map_inline:nn
1854     { underdot , underline , underdblline , underwave , sout , xout }
1855     { \__ctex_clear_fntef_color:n {#1} }
1856 }
1857 </pdfTeX|xetex>

```

14.3.9 \ccwd 的更新

```

\ctex_update_ccwd: 1858 \cs_new_protected:Npn \ctex_update_ccwd:
\ccwd 1859 <*pdfTeX|xetex>
1860 {
1861     \hbox_set:Nn \l__ctex_tmp_box { \CJKglue }
1862     \dim_set:Nn \ccwd { \box_wd:N \l__ctex_tmp_box + \f@size \p@ }
1863 }
1864 </pdfTeX|xetex>
1865 <*luatex>
1866 { \skip_set:Nn \ccwd { \ltjgetparameter { kanjiskip } + \zw } }
1867 </luatex>
1868 <*uptex|aptex>
1869 { \skip_set:Nn \ccwd { 1zw + \tex_kanjiskip:D } }
1870 </uptex|aptex>
1871 \dim_new:N \ccwd

```

\ctex_update_ccglue: 更新字间距。

```

1872 \cs_new_protected:Npn \ctex_update_ccglue:
1873 <*pdfTeX|xetex>
1874 {
1875     \cs_set_protected:Npn \CJKglue
1876     { \skip_horizontal:N \l__ctex_ccglue_skip }
1877 }
1878 </pdfTeX|xetex>
1879 <*luatex>
1880 { \ctex_ltj_set_kanjiskip:N \l__ctex_ccglue_skip }
1881 </luatex>
1882 <*uptex|aptex>
1883 { \skip_set_eq:NN \tex_kanjiskip:D \l__ctex_ccglue_skip }
1884 </uptex|aptex>
1885 \skip_new:N \l__ctex_ccglue_skip

```

\ctex_if_ccglue_touched_p: 检查用户是否修改过汉字间距。

\ctex_if_ccglue_touched: TF

```

1886 \prg_new_conditional:Npnn \ctex_if_ccglue_touched: { TF }
1887 {
1888 <*pdfTeX|xetex>
1889     \if_meaning:w \CJKglue \__ctex_ccglue:
1890     \prg_return_false: \else: \prg_return_true: \fi:
1891 </pdfTeX|xetex>
1892 <*luatex>
1893     \skip_if_eq:nnTF { \l__ctex_ccglue_skip } { \ltjgetparameter { kanjiskip } }
1894     { \prg_return_false: } { \prg_return_true: }
1895 </luatex>
1896 <*uptex|aptex>
1897     \skip_if_eq:nnTF { \l__ctex_ccglue_skip } { \tex_kanjiskip:D }
1898     { \prg_return_false: } { \prg_return_true: }
1899 </uptex|aptex>
1900 }
<pdfTeX|xetex> 1901 \ctex_at_end:n { \cs_new_eq:NN \__ctex_ccglue: \CJKglue }

```

`\ctex_update_em_unit:` 将当前汉字的宽度保存到 `\ccwd` 中备用。不采用 `1em`, 因为这时的 `1em` 实际上来自西文字体的信息, 未必等于汉字的宽度, 这似乎在传统的 `.tfm` 字体上表现更明显。在 `pdfTeX` 和 `XYTeX` 下, 直接使用 `\f@size\p@` 作为汉字的宽度, 这应该对大多数汉字字体都成立, 但不适用于诸如“方正兰亭黑长”之类的特殊字体。在 `XYTeX` 可以用 `\fontcharwd` 来改进。而在 `pdfTeX` 下, 若使用 `zhmetrics` 技术, 所有的汉字共享同一个 `.tfm`, `\fontcharwd` 也就没有意义。在 `LuaTeX` 下, `LuaTeX-jā` 总是按照 JFM 中的设置输出汉字的宽度, 可以直接用 `\zw` 作为汉字宽度。`upTeX` 可以直接使用原生的长度单位 `zw`。

```

1902 \cs_new_protected:Npn \ctex_update_em_unit:
<pdfTeX|xetex> 1903 { \dim_set:Nn \ccwd { \f@size \p@ } }
<luatex> 1904 { \dim_set:Nn \ccwd { \zw } }
<uptex|aptex> 1905 { \dim_set:Nn \ccwd { 1zw } }

```

14.3.10 其他

`\ctex_add_to_selectfont:n` `\CTEX@selectfont@hook` `\EverySelectfont` 直到文档开始时才有效。为了 `\ccwd` 和 `LuaTeX-jā` 的字体设置在导言区也可用, 我们还需要在这里手工修改 `\selectfont`。`everyysel` 宏包会用 `\CheckCommand` 来检查 `\selectfont` 是否为标准定义。我们修改了 `\selectfont`, 所以会给出一个警告。为了消除这个警告, 在它检查之前, 还原本来定义。`pxeveryysel` 宏包取消了检查, 但也需要恢复定义, 避免重复使用钩子。`KOMA-Script` 宏包也会进行检查, 我们需要小心处理。

```

1906 \cs_new_protected:Npn \ctex_add_to_selectfont:n #1
1907 {
1908   \cs_set_protected:Npx \CTEX@selectfont@hook
1909   { \exp_not:o { \CTEX@selectfont@hook #1 } }
1910 }
1911 \cs_new_eq:NN \CTEX@selectfont@hook \prg_do_nothing:
1912 \if_cs_exist:N \@EverySelectfont@Init
1913 \group_begin:
1914   \cs_set:Npn \__ctex_tmp:N #1
1915   {
1916     \tl_set:Nn \l__ctex_tmp_tl {#1}
1917     \cs_new_eq:NN \CTEX@selectfont@save #1
1918     \cs_new_protected:Npn \__ctex_restore_selectfont:
1919     {
1920       \cs_if_free:NF \scr@new@selectfont
1921       {

```

CJK 直接修改 `\selectfont` 和 `pxeveryysel` 的补丁, 会使 `KOMA-Script` 的 `\par@update` 失效。

```

1922 <*pdfTeX|uptex|aptex>
1923   \cs_if_free:NF \par@update
1924 <*uptex|aptex>
1925   {
1926     \tl_put_right:Nn \@EverySelectfont@Init
1927     { \tl_put_right:Nn #1 { \par@update } }
1928   }
1929 </uptex|aptex>
1930 <*pdfTeX>
1931   { \tl_put_right:Nn #1 { \par@update } }
1932   \cs_set_eq:NN \scr@selectfont \CTEX@selectfont@save
1933 </pdfTeX>
1934 </pdfTeX|uptex|aptex>
1935   \cs_set_eq:NN \scr@new@selectfont #1
<!pdfTeX> 1936   \cs_set_eq:NN \CTEX@selectfont@save \scr@selectfont
1937   }
1938   \tl_put_left:Nn \@EverySelectfont@Init
1939   { \cs_set_eq:NN #1 \CTEX@selectfont@save }
1940   \cs_undefine:N \__ctex_restore_selectfont:
1941 }
1942 }
1943 \ctex_parse_name:NN \__ctex_tmp:N \selectfont
1944 \exp_last_unbraced:NNo \group_end:

```



```

1945 \ctex_patch_cmd_once:NnnnTF { \l__ctex_tmp_tl }
1946 { \ExplSyntaxOff }
1947 { \size@update }
1948 { \CTEX@selectfont@hook \size@update }
1949 { \__ctex_restore_selectfont: }
1950 { \ctex_patch_failure:N \selectfont }
1951 \fi:

```

\CJK@plane 有定义,说明处于 CJK 宏包的 \CJKsymbol 之内,不必使用钩子。

```

1952 <*pdfTeX>
1953 \EverySelectfont { \cs_if_exist:NF \CJK@plane { \CTEX@selectfont@hook } }
1954 </pdfTeX>
1955 <*xetex|luatex|uptex|aptex>
1956 \EverySelectfont { \CTEX@selectfont@hook }
1957 </xetex|luatex|uptex|aptex>

```

Attribute 寄存器 \ltj@curjfont 的初始值是 -1,必须把它设置为一个有效的 font.id,否则编译时会直接退出。

```

1958 <*luatex>
1959 \ctex_add_to_selectfont:n
1960 {
1961   \ctex_ltj_select_font:
1962   \ctex_ltj_select_alternate_font:
1963 }
1964 \tl_set:Nn \CJK@family { song } \selectfont
1965 \tl_clear:N \CJK@family
1966 </luatex>

```

\ctex_update_xkanjiskip:
\l__ctex_xkanjiskip_skip

upTeX 和 LuaTeX-jā 对 \xkanjiskip 都是即时赋值。单位 zw 与字体相关,因此需要每次 \selectfont 的时候更新一次 \xkanjiskip。如果用户设置过 \xkanjiskip,就不更新。注意,同 T_EX 的 \baselineskip 一样,如果在一个段落内多次设置了 \kanjiskip 或 \xkanjiskip,只有最后的设置会影响全段。

```

1967 <*luatex|uptex|aptex>
1968 \cs_new_protected:Npn \ctex_update_xkanjiskip:
1969 {
1970   \skip_if_eq:nnT
1971   { \ltjgetparameter { xkanjiskip } } { \l__ctex_xkanjiskip_skip }
1972   { \tex_xkanjiskip:D } { \l__ctex_xkanjiskip_skip }
1973   {
1974     \skip_set:Nn \l__ctex_xkanjiskip_skip { \l__ctex_xkanjiskip_tl }
1975     \ctex_ltj_set_xkanjiskip:N \l__ctex_xkanjiskip_skip
1976     \skip_set_eq:NN \tex_xkanjiskip:D \l__ctex_xkanjiskip_skip
1977   }
1978 }
1979 \tl_new:N \l__ctex_xkanjiskip_tl
1980 \tl_set:Nn \l__ctex_xkanjiskip_tl
1981 { .25\zw plus 1pt minus 1pt }
1982 { .25zw plus 1pt minus 1pt }
1983 \skip_new:N \l__ctex_xkanjiskip_skip
1984 \skip_set:Nn \l__ctex_xkanjiskip_skip
1985 { \ltjgetparameter { xkanjiskip } }
1986 { \tex_xkanjiskip:D }

1987 \ctex_add_to_selectfont:n { \ctex_update_xkanjiskip: }
1988 </luatex|uptex|aptex>

```

\cht 分别从 .jfm 中读取字符高度、深度和宽度,目前仅考虑横排的情况。

```

\cdp
\cwd
1989 <*luatex>
1990 \dim_new:N \cht
1991 \dim_new:N \cdp
1992 \dim_new:N \cwd
\ctex_update_kanjisize:

```

```

1993 \group_begin:
1994 \char_set_catcode_space:n { 32 }
1995 \lua_now:e
1996 {
1997     local nulltable = { }
1998     local fmt = luatexja.jfont.font_metric_table
1999     local getattribute = tex.getattribute
2000     local setdimen = tex.setdimen
2001     local func = lua.get_functions_table()
2002     local id = luatexbase.new_luafunction("ctex")
2003     func[id] = function ()
2004         local ft = fmt[getattribute("ltj@curjfont")] or nulltable
2005         local ft = ft and ft.char_type or nulltable
2006         local fk = ft and ft[0] or nulltable
2007         setdimen("cht", fk.height or 0)
2008         setdimen("cdp", fk.depth or 0)
2009         setdimen("cwd", fk.width or ft.zw or 0)
2010     end
2011     token.set_lua("ctex_update_kanjisize:", id, "global", "protected")
2012 }
2013 \group_end:
2014 \ctex_add_to_selectfont:n { \ctex_update_kanjisize: }
2015 </luatex>

```

space 在导言区或正文中设置忽略空格方式。pdfTeX 和 XeTeX 下初始设置为 auto, LuaTeX、upTeX 下是无效选项。

```

2016 \ctex_define:n
2017 {
2018     <pdf|xetex>
2019     space .choice: ,
2020     space / true .code:n =
2021     { \ctex_ignorespaces_case:N \prg_do_nothing: } ,
2022     <xetex> { \xeCJKsetup { CJKspace = true } } ,
2023     space / auto .code:n =
2024     { \ctex_ignorespaces_case:N \ctex_auto_ignorespaces: } ,
2025     <xetex> { \xeCJKsetup { CJKspace = false } } ,
2026     space / false .code:n =
2027     { \ctex_ignorespaces_case:N \tex_ignorespaces:D } ,
2028     <xetex> { \xeCJKsetup { CJKspace = false } } ,
2029     space .default:n = { true } ,
2030     space .initial:n = { auto }
2031 </pdf|xetex>
2032 <lua|uptex|aptex>
2033     space .code:n =
2034     { \msg_warning:nn { ctex } { invalid-option } }
2035 </lua|uptex|aptex>
2036 }

```

punct 在导言区或正文中设置标点符号输出格式。LuaTeX-jaf 设置的是字体的默认 JFM, 只会影响到之后设置的字体。upTeX 暂时无效。

```

2037 \ctex_define:n
2038 {
2039     punct .code:n =
2040     {
2041         \tl_set:Nx \l__ctex_punct_tl {#1}
2042         \punctstyle { \l__ctex_punct_tl }
2043         <xetex> \xeCJKsetup { PunctStyle = \l__ctex_punct_tl }
2044         <lua> \ctex_mono_jfm:o { \l__ctex_punct_tl }
2045         <uptex|aptex> \msg_warning:nn { ctex } { invalid-option }
2046     } ,
2047     punct .default:n = { quanjiao } ,
2048 }
2049 </pdf|xetex|lua|uptex|aptex>

```

14.3.11 载入引擎定义文件

最后载入各个编译引擎的定义文件。

```
<class|ctex> 2050 \ctex_file_input:n { \c__ctex_engine_file_str }
```

14.4 用户设置接口

```
\ctexset 2051 <*class|ctex|ctexheading>
2052 \NewDocumentCommand \ctexset { } { \ctex_set:n }
2053 </class|ctex|ctexheading>
```

\CTEXsetup
\CTEXoptions 过时命令。

```
2054 <*class|ctex>
2055 \NewDocumentCommand \CTEXsetup { +0 { } } > { \TrimSpaces } m }
2056 {
2057   \tl_if_blank:nTF {#1}
2058   { \ctex_deprecated_command:Nn \CTEXsetup { } }
2059   {
2060     \ctex_deprecated_command:Nn \CTEXsetup
2061     { \ctexset {~#2~{~#1~}~}is~set. }
2062     \ctex_set:nn {#2} {#1}
2063   }
2064 }
2065 \NewDocumentCommand \CTEXoptions { +0 { } }
2066 {
2067   \tl_if_blank:nTF {#1}
2068   { \ctex_deprecated_command:Nn \CTEXoptions { } }
2069   {
2070     \ctex_deprecated_command:Nn \CTEXoptions
2071     { \ctexset {~#1~}is~set. }
2072     \ctex_set:n {#1}
2073   }
2074 }
```

14.5 字距与缩进

autoindent autoindent 也是可以用在正文中的选项,意义与宏包选项 option/autoindent 相同。

```
2075 \ctex_define:n
2076 {
2077   autoindent .choice: ,
2078   autoindent .default:n = { true } ,
2079   autoindent / true .code:n =
2080   {
2081     \tl_set:Nn \l__ctex_autoindent_tl { 2 \ccwd }
2082     \ctex_select_size:
2083   } ,
2084   autoindent / false .code:n =
2085   { \tl_clear:N \l__ctex_autoindent_tl } ,
2086   autoindent / unknown .code:n =
2087   {
2088     \ctex_set_default_ccwd:Nn \l__ctex_autoindent_tl {#1}
2089     \ctex_select_size:
2090   }
2091 }
```

\CTEXsetfont 无论字体大小是否变化都更新相关信息。

```
2092 \NewDocumentCommand \CTEXsetfont { } { \ctex_select_size: }
2093 \cs_new_protected:Npn \ctex_select_size:
2094 { \cs_if_free:NTF \size@update { \ctex_update_size: } { \selectfont } }
```

`\ctex_update_size:` 在字号变化时更新 `\ccwd`、`\parindent` 和汉字间距。字距为零则恢复正常设置。

```

2095 \cs_new_protected:Npn \ctex_update_size:
2096 {
2097   \tl_if_eq:NNTF \l__ctex_ziju_tl \c__ctex_zero_tl
2098   {
2099     \ctex_update_stretch:
2100     \ctex_update_parindent:
2101   }
2102   { \ctex_update_ziju: }
2103 }
2104 \tl_const:Nx \c__ctex_zero_tl { \fp_use:N \c_zero_fp }
2105 \tl_new:N \l__ctex_ziju_tl
2106 \tl_set_eq:NN \l__ctex_ziju_tl \c__ctex_zero_tl

```

在 `\selectfont` 中, 若 `\size@update` 为 `\relax`, 说明字体大小没有变化, 我们也就不用更新相关参数。

```

2107 \ctex_add_to_selectfont:n
2108 { \cs_if_free:NF \size@update { \ctex_update_size: } }

```

linestretch 若行宽不是汉字宽度的整数倍, 自然要求伸展它们之间的差。这里设置的是在此基础上的额外伸展量。初始化为一个汉字的宽度。若设置为 `\maxdimen`, 则禁用此功能。参数的默认单位是汉字的宽度 `\ccwd`。

```

2109 \ctex_define:n
2110 {
2111   linestretch .code:n =
2112   {
2113     \ctex_set_default_ccwd:Nn \l__ctex_line_stretch_tl {#1}
2114     \ctex_select_size:
2115   },
2116   linestretch .value_required:n = true
2117 }
2118 \tl_new:N \l__ctex_line_stretch_tl
2119 \tl_set:Nn \l__ctex_line_stretch_tl { \ccwd }

```

`\ctex_update_stretch:` 首先计算一行上汉字的字数, `\CJKglue` 相当于将 `\linewidth` 与汉字总宽度之差均匀地填充到汉字之间。 ε - $\mathrm{T}_{\mathrm{E}}\mathrm{X}$ 的除法是四舍五入, 而我们这里应该用截断。由于没有可展性的要求, 直接用原语 `\tex_divide:D` 要比 `\int_div_truncate:nn` 快一些。下面的算法还兼顾到了 `\linewidth` 不为汉字字宽的整数倍的情况。若用户禁用 `linestretch` 并且修改过 `\CJKglue`, 则只更新 `\ccwd`, 否则设置伸展量为 0.08 倍 `\baselineskip`。注意 `everysel` 的钩子位于 `\size@update` 之前, `\baselineskip` 还未更新, 不能直接使用它。

```

2120 \cs_new_protected:Npn \ctex_update_stretch:
2121 {
2122   \ctex_update_em_unit:
2123   \dim_set:Nn \l__ctex_tmp_dim { \l__ctex_line_stretch_tl }
2124   \dim_compare:nNnTF \l__ctex_tmp_dim = \c_max_dim
2125   {
2126     \ctex_if_ccglue_touched:TF
2127     { \ctex_update_ccwd: }
2128     {
2129       \dim_set:Nn \l__ctex_tmp_dim
2130       { \baselinestretch \tex_glueexpr:D \f@baselineskip \scan_stop: }
2131       \skip_set:Nn \l__ctex_ccglue_skip
2132       { \c_zero_dim plus .08 \l__ctex_tmp_dim }
2133       \ctex_update_ccglue:
2134     }
2135   }
2136   {
2137     \int_set:Nn \l__ctex_tmp_int
2138     { \tex_dimexpr:D \linewidth - \ccwd - \l__ctex_tmp_dim \scan_stop: }
2139     \tex_divide:D \l__ctex_tmp_int \ccwd

```

```

2140     \int_compare:nNnTF \l__ctex_tmp_int > \c_zero_int
2141     {
2142         \skip_set:Nn \l__ctex_ccglue_skip
2143         {
2144             \c_zero_dim plus \dim_eval:n
2145             {
2146                 ( \linewidth - \ccwd - \l__ctex_tmp_int \ccwd ) /
2147                 \l__ctex_tmp_int
2148             }
2149         }
2150     }
2151     { \skip_zero:N \l__ctex_ccglue_skip }
2152     \ctex_update_ccglue:
2153 }
2154 }

```

`\ctex_update_parindent:` 更新段落首行缩进。此函数在字号变化时调用。

```

2155 \cs_new_protected:Npn \ctex_update_parindent:
2156 {
2157     \tl_if_empty:NF \l__ctex_autoindent_tl
2158     {
2159         \dim_compare:nNnF \parindent = \c_zero_dim
2160         { \dim_set:Nn \parindent { \l__ctex_autoindent_tl } }
2161     }
2162 }

```

`\ziju` 若参数为 0, 则恢复正常间距。

```

2163 \NewDocumentCommand \ziju { m }
2164 { \exp_args:Nx \ctex_ziju:n {#1} \tex_ignorespaces:D }
2165 \cs_new_protected:Npn \ctex_ziju:n #1
2166 {
2167     \tl_set:Nx \l__ctex_ziju_tl { \fp_eval:n {#1} }
2168     \ctex_select_size:
2169 }

```

`\ctex_update_ziju:` 更新字距。若字距不大于 -1 , 即 `\ccwd` 为非正值, 则不计算伸缩值。否则, 首先假定汉字的宽度为正常宽度加上字距, 看一行上能正常放下多少个汉字。

```

2170 \cs_new_protected:Npn \ctex_update_ziju:
2171 {
2172     \ctex_update_em_unit:
2173     \dim_set:Nn \l__ctex_ziju_dim { \l__ctex_ziju_tl \ccwd }
2174     \dim_add:Nn \ccwd { \l__ctex_ziju_dim }
2175     \dim_compare:nNnTF \ccwd > \c_zero_dim

```

伸展量保证行内的剩余空白能够被均匀地填充到汉字之间, 收缩的最大限度是让当前行还能够再挤下一个汉字并且不会出现负间距。由 $\mathrm{T}_{\mathrm{E}}\mathrm{X}$ 决定伸展还是收缩。

```

2176     {
2177         \dim_set:Nn \l__ctex_tmp_dim
2178         { \linewidth - \ccwd + \l__ctex_ziju_dim }
2179         \int_set:Nn \l__ctex_tmp_int { \l__ctex_tmp_dim }
2180         \tex_divide:D \l__ctex_tmp_int \ccwd
2181         \dim_sub:Nn \l__ctex_tmp_dim { \l__ctex_tmp_int \ccwd }

```

由于 `\parindent` 是一个固定值, 并不参与伸缩, 容易导致第一行出现坏盒子。我们在这里将字数减去 2, 以此放大伸缩值。

```

2182         \dim_compare:nNnF \parindent = \c_zero_dim
2183         {
2184             \int_compare:nNnF \l__ctex_tmp_int < 3
2185             { \int_sub:Nn \l__ctex_tmp_int { 2 } }
2186         }
2187         \skip_set:Nn \l__ctex_ccglue_skip

```

```

2188     {
2189         \l__ctex_ziju_dim
2190         plus \dim_eval:n { \l__ctex_tmp_dim / \l__ctex_tmp_int }
2191         minus \dim_min:nn { \dim_abs:n { \l__ctex_ziju_dim } }
2192         { ( \ccwd - \l__ctex_tmp_dim ) / ( \l__ctex_tmp_int + 1 ) }
2193     }
2194 }
2195 { \skip_set:Nn \l__ctex_ccglue_skip { \l__ctex_ziju_dim } }
2196 \ctex_update_ccglue:

```

字距设置得比较大时, 为了尽量保证段首缩进能够与下一行对齐, 应该需要相应地加上或者减去伸缩值。但是这里并不清楚 $\text{T}_{\text{E}}\text{X}$ 是伸展还是收缩, 之前以“当前行是否还放得下一个汉字”为标准加上或减去伸缩值的做法也未必与实际结果一致, 所以只好还是设置为 $2\backslash\text{ccwd}$ 。

```

2197 \ctex_update_parindent:
2198 }
2199 \dim_new:N \l__ctex_ziju_dim

```

$\backslash\text{CTEXindent}$
 $\backslash\text{CTEXnoindent}$

过时命令。

```

2200 \NewDocumentCommand \CTEXindent { }
2201 {
2202     \ctex_deprecated_command:Nn \CTEXindent
2203     { \parindent is~set~to~2\ccwd. }
2204     \ctex_update_ccwd:
2205     \dim_set:Nn \parindent { 2 \ccwd }
2206 }
2207 \NewDocumentCommand \CTEXnoindent { }
2208 {
2209     \ctex_deprecated_command:Nn \CTEXnoindent
2210     { \parindent is~set~to~0pt. }
2211     \dim_zero:N \parindent
2212 }

```

14.6 中文数字与日期

```

2213 \PassOptionsToPackage { encoding = \g__ctex_encoding_tl } { zhnumber }
2214 \RequirePackage { zhnumber }

```

$\backslash\text{chinese}$

```

2215 \cs_new:Npn \chinese { \zhnum_counter:n }
2216 \cs_new_eq:NN \@chinese \@zhnum
2217 \cs_new_eq:NN \Chinese \chinese
2218 \cs_new_eq:NN \CTEXcounter \use_none:n

```

给 `enumitem` 宏包注册 $\backslash\text{chinese}$ 、 $\backslash\text{zhnum}$ 和 $\backslash\text{zhdig}$ 。

```

2219 \ctex_at_end_package:nn { enumitem }
2220 {
2221     \cs_if_free:NF \AddEnumerateCounter
2222     {
2223         \AddEnumerateCounter * { \zhnum } { \@zhnum } { 1 }
2224         \AddEnumerateCounter * { \zhdig } { \@zhdig } { 1 }
2225         \AddEnumerateCounter * { \chinese } { \@chinese } { 1 }
2226     }
2227 }

```

$\backslash\text{CTEXnumber}$
 $\backslash\text{CTEXdigits}$

```

2228 \NewDocumentCommand \CTEXnumber { m m }
2229 { \protected@edef #1 { \zhnumber {#2} } }
2230 \NewDocumentCommand \CTEXdigits { m m }
2231 { \protected@edef #1 { \zhdigits {#2} } }

```

today

```

2232 \cs_set_eq:NN \CTEX@todayold \today
2233 \ctex_define:n
2234 {
2235     today .choice: ,

```

```

2236 today / old .code:n =
2237 { \cs_set_eq:NN \today \CTEX@todayold } ,
2238 today / small .code:n =
2239 {
2240 \cs_set_eq:NN \today \zhtoday
2241 \zhnumsetup { time = Arabic }
2242 } ,
2243 today / big .code:n =
2244 {
2245 \cs_set_eq:NN \today \zhtoday
2246 \zhnumsetup { time = Chinese }
2247 } ,
2248 today / unknown .code:n =
2249 { \msg_error:nxx { ctex } { today-undef } {#1} }
2250 }
2251 \msg_new:nnnn { ctex } { today-undef }
2252 { Today~format~`#1'~is~undefined. }
2253 { Available~today~formats~are~`old',~`small',~and~`big'. }

```

14.7 其他中文标题定义

`\proofname` `\proofname` 未在标准文档类中定义, 需要确保它非空。

```

2254 \tl_if_exist:NF \proofname
2255 {
2256 \tl_new:N \proofname
2257 \tl_set:Nn \proofname { Proof }
2258 }

2259 \ctex_define:n
2260 {
2261 contentsname .tl_set:N = \contentsname ,
2262 listfigurename .tl_set:N = \listfigurename ,
2263 listtablename .tl_set:N = \listtablename ,
2264 figurename .tl_set:N = \figurename ,
2265 tablename .tl_set:N = \tablename ,
2266 abstractname .tl_set:N = \abstractname ,
2267 indexname .tl_set:N = \indexname ,
2268 appendixname .tl_set:N = \appendixname ,
2269 proofname .tl_set:N = \proofname ,
2270 bibname .tl_set:N = \refname
2271 bibname .tl_set:N = \bibname
2272 }
2273 algorithmname .tl_set:N = \algorithmname ,
2274 bibname .tl_set:N = \bibname ,
2275 refname .tl_set:N = \refname ,
2276 continuation .tl_set:N = \insertcontinuationtext
2277 }
2278 }

2279 }
2280 \msg_new:nnn { ctex } { ctexbibname }
2281 {
2282 Neither~`\token_to_str:N \bibname'~nor~`\token_to_str:N \refname'~can~be~found.~\
2283 The~key~`\bibname'~will~set~`\token_to_str:N \ctexbibname'~to~the~given~value.
2284 }
2285 \tl_if_exist:NTF \insertcontinuationtext
2286 {
2287 \ctex_define:n
2288 {
2289 algorithmname .tl_set:N = \algorithmname ,
2290 bibname .tl_set:N = \bibname ,
2291 refname .tl_set:N = \refname ,
2292 continuation .tl_set:N = \insertcontinuationtext
2293 }
2294 }
2295 {

```

```

2296 \tl_if_exist:NTF \bibname
2297 { \ctex_define:n { bibname .tl_set:N = \bibname } }
2298 {
2299 \tl_if_exist:NTF \refname
2300 { \ctex_define:n { bibname .tl_set:N = \refname } }
2301 {
2302 \msg_warning:nn { ctex } { ctexbibname }
2303 \ctex_define:n { bibname .tl_set:N = \ctexbibname }
2304 }
2305 }
2306 }
2307 </ctex>
2308 </class|ctex>

```

14.8 中文化的标题结构

本节内容在 C_TE_X 文档类或打开 heading 选项下生效。

```
2309 <*class|heading>
```

14.8.1 定义标题格式选项

\c__ctex_section_headings_seq 保存 \section 级以下标题名字。

```

2310 <*article|book|report>
2311 \seq_const_from_clist:Nn \c__ctex_section_headings_seq
2312 { section , subsection , subsubsection , paragraph , subparagraph }
2313 </article|book|report>

```

```

\c__ctex_headings_seq 2314 <*article|book|report>
2315 \seq_new:N \c__ctex_headings_seq
2316 \seq_gset_eq:NN \c__ctex_headings_seq \c__ctex_section_headings_seq
<book|report> 2317 \seq_gput_left:Nn \c__ctex_headings_seq { chapter }
2318 \seq_gput_left:Nn \c__ctex_headings_seq { part }
2319 </article|book|report>
2320 <*beamer>
2321 \seq_const_from_clist:Nn \c__ctex_headings_seq
2322 { part , section , subsection }
2323 </beamer>

```

```

\__ctex_initial_heading:n 2324 \cs_new_protected:Npn \__ctex_initial_heading:n #1
2325 {
2326 \tl_new:c { CTEX@pre#1 }
2327 \tl_new:c { CTEX@post#1 }
2328 \tl_const:cx { CTEX@the#1 }
2329 {
2330 \exp_not:c { CTEX@pre#1 }
2331 \exp_not:c { CTEX@the#1 }
2332 \exp_not:c { CTEX@post#1 }
2333 }
2334 \tl_const:cx { CTEX@#1name }
2335 {
2336 \group_begin:
2337 \exp_not:c { CTEX@#1@nameformat }
2338 {
2339 \exp_not:c { CTEX@pre#1 }
2340 \exp_not:N \tl_if_empty:NTF
2341 \exp_not:c { CTEX@#1@numberformat }
2342 { \exp_not:c { CTEX@the#1 } }
2343 {
2344 \group_begin:
2345 \exp_not:c { CTEX@#1@numberformat }
2346 \exp_not:c { CTEX@the#1 }

```



```

2347         \group_end:
2348     }
2349     \exp_not:c { CTEX@post#1 }
2350 }
2351 \group_end:
2352 }
2353 }

\__ctex_def_heading_keys:n 2354 \cs_new_protected:Npn \__ctex_def_heading_keys:n #1
2355 {
2356     \tl_put_right:Nx \l__ctex_tmp_tl
2357     {
2358         #1 .meta:nn = { ctex / #1 } { #####1 } ,
2359         #1 / name .code:n =
2360         { \ctex_assign_heading_name:nn {#1} { #####1 } } ,
2361         #1 / number .tl_set:N = \exp_not:c { CTEX@the#1 } ,
2362         #1 / beforekip .tl_set:N = \exp_not:c { CTEX@#1@beforekip } ,
2363         #1 / afterskip .tl_set:N = \exp_not:c { CTEX@#1@afterskip } ,
2364         #1 / indent .tl_set:N = \exp_not:c { CTEX@#1@indent } ,
2365         #1 / numbering .bool_set:N = \exp_not:c { CTEX@#1@numbering } ,
2366         #1 / numbering .initial:n = true ,
2367         #1 / beforekip .initial:n = \c_zero_skip ,
2368         #1 / afterskip .initial:n = \c_zero_skip ,
2369         #1 / indent .initial:n = \c_zero_dim ,
2370         #1 / beforekip .value_required:n = true ,
2371         #1 / afterskip .value_required:n = true ,
2372         #1 / indent .value_required:n = true ,
2373         < *article|book|report >
2374         #1 / afterindent .bool_set:N = \exp_not:c { CTEX@#1@afterindent } ,
2375         #1 / fixskip .bool_set:N = \exp_not:c { CTEX@#1@fixskip } ,
2376         #1 / hang .bool_set:N = \exp_not:c { CTEX@#1@hang } ,
2377         #1 / hang .initial:n = true ,
2378         #1 / runin .bool_set:N = \exp_not:c { CTEX@#1@runin } ,
2379         #1 / tocline .cs_set:Np = \exp_not:c { CTEX@#1@tocline }
2380         \exp_not:n { ##1##2 } ,
2381         \__ctex_plus_key_aux:nn {#1} { break } ,
2382         < /article|book|report >
2383         \__ctex_plus_key_aux:nn {#1} { format } ,
2384         \__ctex_plus_key_aux:nn {#1} { nameformat } ,
2385         \__ctex_plus_key_aux:nn {#1} { numberformat } ,
2386         \__ctex_plus_key_aux:nn {#1} { titleformat } ,
2387         \__ctex_plus_key_aux:nn {#1} { aftername } ,
2388         \__ctex_plus_key_aux:nn {#1} { aftertitle } ,
2389     }
2390 }
2391 \cs_new:Npn \__ctex_plus_key_aux:nn #1#2
2392 {
2393     #1 / #2 .tl_set:N = \exp_not:c { CTEX@#1@#2 } ,
2394     #1 / #2 + .code:n =
2395     { \tl_put_right:Nn \exp_not:c { CTEX@#1@#2 } { #####1 } } ,
2396     #1 / #2 ~ + .code:n =
2397     { \tl_put_right:Nn \exp_not:c { CTEX@#1@#2 } { #####1 } }
2398 }

```

`\ctex_assign_heading_name:nn` 的 `name` 的值是一个至多两个元素的逗号分隔列表。由于 \LaTeX 3 的 `clist` 总是会自动忽略空元素，所以设置 `name={, 章}` 后，第一个元素将会是“章”，必须用空的分组保护空元素：`name={ {}, 章 }`，这在使用中有些许不便。我们可以改用 `seq` 或者手写函数解析参数来加以改进。为实现的简单起见，这里用了 `xparse` 的 `\SplitArgument`，它带有参数的长度检查。

```

2399 \NewDocumentCommand \ctex_assign_heading_name:nn
2400 { m > { \SplitArgument { 1 } { , } } +m }
2401 { \__ctex_assign_heading_name:nnn {#1} #2 }
2402 \cs_new_protected:Npn \__ctex_assign_heading_name:nnn #1#2#3
2403 {
2404     \tl_set:cn { CTEX@pre#1 } {#2}

```

```

2405 \tl_if_novalue:nTF {#3}
2406 { \tl_clear:c { CTEX@post#1 } }
2407 { \tl_set:cn { CTEX@post#1 } {#3} }
2408 }

```

part/pagestyle
chapter/pagestyle
chapter/lofskip
chapter/lotskip

只在 `ctexbook` 和 `ctexrep` 下有定义。

```

2409 \group_begin:
2410 <*book|report>
2411 \tl_set:Nn \l__ctex_tmp_tl
2412 {
2413   part / pagestyle .tl_set:N = \CTEX@part@pagestyle ,
2414   chapter / pagestyle .tl_set:N = \CTEX@chapter@pagestyle ,
2415   chapter / lofskip .tl_set:N = \CTEX@chapter@lofskip ,
2416   chapter / lotskip .tl_set:N = \CTEX@chapter@lotskip ,
2417   chapter / lofskip .initial:n = \c_zero_skip ,
2418   chapter / lotskip .initial:n = \c_zero_skip ,
2419   chapter / lofskip .value_required:n = true ,
2420   chapter / lotskip .value_required:n = true ,
2421 }
2422 </book|report>
2423 <*article|beamer>
2424 \tl_clear:N \l__ctex_tmp_tl
2425 </article|beamer>

```

定义标题键值选项。

```

2426 \seq_map_inline:Nn \c__ctex_headings_seq
2427 {
2428   \__ctex_initial_heading:n {#1}
2429   \__ctex_def_heading_keys:n {#1}
2430 }
2431 \exp_args:NNo \group_end: \ctex_define:n { \l__ctex_tmp_tl }

```

\CTEX@heading@format@initial

标题格式的一些初始设置，包括恢复默认字体，并禁用自动调整首行缩进，禁止在标题中分页。同时用 `\noindent` 抑制首行缩进并进入水平模式。统一在各级标题的 `format` 选项之前使用。

```

2432 \cs_new_protected:Npn \CTEX@heading@format@initial
2433 {
2434   \normalfont
2435   \tl_clear:N \l__ctex_autoindent_tl
2436   \int_set:Nn \tex_interlinepenalty:D { 10 000 }
2437   \tex_noindent:D
2438 }

```

\ctex_indent_box:n

设置 `\parindent`，并插入用于产生缩进的盒子，如果缩进为 0，就不插入。

```

2439 \cs_new_protected:Npn \ctex_indent_box:n #1
2440 {
2441   \dim_set:Nn \tex_parindent:D {#1}
2442   \__ctex_insert_indent:
2443 }
2444 \cs_new_protected:Npn \__ctex_insert_indent:
2445 {
2446   \dim_compare:nNnF \tex_parindent:D = \c_zero_dim
2447   { \tex_indent:D }
2448 }
2449 \cs_new_eq:NN \CTEX@indentbox \ctex_indent_box:n

```

14.8.2 标准标题命令的修改

```

2450 <*article|book|report>

```

`\CTEX@fixtopskip` 修正 `book` 和 `report` 类的 `\part` 和 `\chapter` 标题之前的多余空行。

```

2451 <*book|report>
2452 \cs_new_protected:Npn \CTEX@fixtopskip
2453 {
2454   \CTEX@fixheadingskip
2455   \dim_compare:nNnF \tex_pagegoal:D < \c_max_dim
2456     { \skip_sub:Nn \l__ctex_heading_skip { \tex_topskip:D } }
2457 }
2458 </book|report>

```

`\CTEX@fixheadingskip` 抑制行间粘连,修正标题前后的多余间距。事实上,减掉 `\parskip`,有一定的风险。如果接下来的内容不会进入水平模式(例如在 `format` 选项中使用 `\hrule` 或者 `\hbox`), $\mathrm{T}_{\mathrm{E}}\mathrm{X}$ 就不会加上 `\parskip`。这时候就需要用户把 `\parskip` 加到 `before skip` 或者 `after skip` 作为修正。

```

2459 \cs_new_protected:Npn \CTEX@fixheadingskip
2460 {
2461   \par
2462   \dim_set:Nn \tex_prevdepth:D { -1000pt }
2463   \skip_sub:Nn \l__ctex_heading_skip { \tex_parskip:D }
2464 }
2465 \skip_new:N \l__ctex_heading_skip
2466 \cs_new_protected:Npn \CTEX@setheadingskip
2467 { \skip_set:Nn \l__ctex_heading_skip }
2468 \cs_new_eq:NN \CTEX@headingskip \l__ctex_heading_skip

```

`\partmark` 提供 `\partmark`。

```

2469 \ProvideDocumentCommand \partmark { m }
2470 { \markboth { } { } }

```

`\CTEXifname` 用于判断当前标题是否有编号。

```

\CTEX@ifnametrue
\CTEX@ifnamefalse
2471 \cs_new_eq:NN \CTEXifname \use_ii:nn
2472 \cs_new_protected:Npn \CTEX@ifnametrue
2473 { \cs_set_eq:NN \CTEXifname \use_i:nn }
2474 \cs_new_protected:Npn \CTEX@ifnamefalse
2475 { \cs_set_eq:NN \CTEXifname \use_ii:nn }

```

`\CTEX@addloflotskip` 往插图和表格目录中加入额外间距。如果间距为零,则不加入。

```

2476 <*book|report>
2477 \cs_new_protected:Npn \CTEX@addloflotskip #1
2478 {
2479   \skip_set:Nn \l__ctex_heading_skip { \use:c { CTEX@#1@lofskip } }
2480   \skip_if_eq:nnF { \l__ctex_heading_skip } { \c_zero_skip }
2481   {
2482     \addtocontents { lof }
2483     { \protect \addvspace { \skip_use:N \l__ctex_heading_skip } }
2484   }
2485   \skip_set:Nn \l__ctex_heading_skip { \use:c { CTEX@#1@lotskip } }
2486   \skip_if_eq:nnF { \l__ctex_heading_skip } { \c_zero_skip }
2487   {
2488     \addtocontents { lot }
2489     { \protect \addvspace { \skip_use:N \l__ctex_heading_skip } }
2490   }
2491 }
2492 </book|report>

```

```

\CTEX@addtocline
2493 \cs_new_protected:Npn \CTEX@addtocline #1#2
2494 { \addcontentsline { toc } {#1} { \use:c { CTEX@#1@tocline } {#1} {#2} } }

```

14.8.2.1 part 的标题

```

\part 2495 <*article>
2496 \renewcommand\part{%
2497 \if@noskipsec \leavevmode \fi
2498 \par
2499 \CTEX@part@break
2500 % \addvspace{4ex}%
2501 \CTEX@setheadingskip \CTEX@part@beforeskip
2502 \ifodd \CTEX@part@fixskip \CTEX@fixheadingskip \fi
2503 \addvspace \CTEX@headingskip
2504 \ifodd \CTEX@part@afterindent
2505 \@afterindenttrue
2506 \else
2507 \@afterindentfalse
2508 \fi
2509 \secdef\@part\@spart}
2510 </article>
2511 <*book|report>
2512 \renewcommand\part{%
2513 % \if@openright
2514 % \cleardoublepage
2515 % \else
2516 % \clearpage
2517 % \fi
2518 \CTEX@part@break
2519 % \thispagestyle{plain}%
2520 \thispagestyle{\CTEX@part@pagestyle}%
2521 \if@twocolumn
2522 \onecolumn
2523 \@tempwatrue
2524 \else
2525 \@tempwafalse
2526 \fi
2527 % \null\vfil
2528 \CTEX@setheadingskip \CTEX@part@beforeskip
2529 \ifodd \CTEX@part@fixskip \CTEX@fixtopskip \fi
2530 \vspace*{\CTEX@headingskip}%
2531 \secdef\@part\@spart}
2532 </book|report>

\@part 2533 <*article>
2534 \def\@part[#1]#2{%
2535 \ifnum \c@secnumdepth >\m@ne
2536 \ifodd \CTEX@part@numbering
2537 \CTEX@ifnametrue
2538 \refstepcounter{part}%
2539 % \addcontentsline{toc}{part}{\thepart\hspace{1em}#1}%
2540 \else
2541 \CTEX@ifnamefalse
2542 \CTEX@makeanchor{part*}%
2543 % \addcontentsline{toc}{part}{#1}%
2544 \fi
2545 \else
2546 \CTEX@ifnamefalse
2547 \CTEX@makeanchor{part*}%
2548 % \addcontentsline{toc}{part}{#1}%
2549 \fi
2550 \CTEX@getttitle{#1}%
2551 \CTEX@addtocline{part}{#1}%
2552 \partmark{#1}%
2553 \begingroup
2554 % \parindent \z@ \raggedright \interlinepenalty \@M \normalfont
2555 \CTEX@heading@format@initial
2556 \CTEX@part@format{%
2557 % \ifnum \c@secnumdepth >\m@ne
2558 % \Large\bfseries\partname\nobreakspace\thepart\par\nobreak

```

```

2559 % \fi
2560 \CTEX@headinghang{part}%
2561 {\CTEX@ifname{\CTEX@partname\CTEX@part@aftername}{}}%
2562 % \huge\bfseries #2%
2563 \CTEX@part@titleformat{#2}%
2564 % \markboth{}{}\par
2565 \CTEX@part@aftertitle}\par
2566 \endgroup
2567 \nobreak
2568 % \vskip 3ex
2569 \CTEX@setheadingskip \CTEX@part@afterskip
2570 \ifodd \CTEX@part@fixskip \CTEX@fixheadingskip \fi
2571 \vskip \CTEX@headingskip
2572 \@afterheading}
2573 </article>
2574 <*book|report>
2575 \def\@part[#1]#2{%
2576 \ifnum \c@secnumdepth >-2\relax
2577 \ifodd \CTEX@part@numbering
2578 \CTEX@ifnametrue
2579 \refstepcounter{part}%
2580 % \addcontentsline{toc}{part}{\thepart\hspace{1em}#1}%
2581 \else
2582 \CTEX@ifnamefalse
2583 \CTEX@makeanchor{part*}%
2584 % \addcontentsline{toc}{part}{#1}%
2585 \fi
2586 \else
2587 \CTEX@ifnamefalse
2588 \CTEX@makeanchor{part*}%
2589 % \addcontentsline{toc}{part}{#1}%
2590 \fi
2591 \CTEX@getttitle{#1}%
2592 \CTEX@addtocline{part}{#1}%
2593 % \markboth{}{}\par
2594 \partmark{#1}%
2595 \begingroup
2596 % \centering \interlinepenalty \@M \normalfont
2597 \CTEX@heading@format@initial
2598 \CTEX@part@format{%
2599 % \ifnum \c@secnumdepth >-2\relax
2600 % \huge\bfseries\partname\nobreakspace\thepart\par\vskip 20\p@
2601 % \fi
2602 \CTEX@headinghang{part}%
2603 {\CTEX@ifname{\CTEX@partname\CTEX@part@aftername}{}}%
2604 % \Huge\bfseries #2\par
2605 \CTEX@part@titleformat{#2}%
2606 \CTEX@part@aftertitle}\par
2607 \endgroup
2608 \@endpart}
2609 </book|report>

\@spart 2610 <*article>
2611 \def\@spart#1{%
2612 \CTEX@ifnamefalse
2613 \CTEX@makeanchor@spart{part*}%
2614 \CTEX@getttitle{#1}%
2615 \begingroup
2616 % \parindent \z@ \raggedright \interlinepenalty \@M \normalfont
2617 \CTEX@heading@format@initial
2618 \CTEX@part@format{%
2619 \CTEX@headinghang{part}{}}%
2620 % \huge \bfseries #1\par
2621 \CTEX@part@titleformat{#1}%
2622 \CTEX@part@aftertitle}\par
2623 \endgroup
2624 \nobreak

```

```

2625 % \vskip 3ex
2626 \CTEX@setheadingskip \CTEX@part@afterskip
2627 \ifodd \CTEX@part@fixskip \CTEX@fixheadingskip \fi
2628 \vskip \CTEX@headingskip
2629 \@afterheading}
2630 </article>
2631 <*book|report>
2632 \def\@spart#1{%
2633   \CTEX@ifnamefalse
2634   \CTEX@makeanchor@spart{part*}%
2635   \CTEX@getttitle{#1}%
2636   \begingroup
2637   % \centering \interlinepenalty \@M \normalfont
2638   \CTEX@heading@format@initial
2639   \CTEX@part@format{%
2640     \CTEX@headinghang{part}{}%
2641   % \Huge \bfseries #1\par%
2642     \CTEX@part@titleformat{#1}%
2643     \CTEX@part@aftertitle}\par
2644   \endgroup
2645   \@endpart}
2646 </book|report>

\@endpart 2647 <*book|report>
2648 \def\@endpart{%
2649 % \vfil
2650 \CTEX@setheadingskip \CTEX@part@afterskip
2651 \ifodd \CTEX@part@fixskip \CTEX@fixheadingskip \fi
2652 \vskip \CTEX@headingskip
2653 \newpage
2654 \if@twoside
2655   \if@openright
2656     \null
2657     \thispagestyle{empty}%
2658   \newpage
2659   \fi
2660 \fi
2661 \if@tempwa
2662   \twocolumn
2663 \fi}
2664 </book|report>

```

14.8.2.2 chapter 的标题

```

2665 <*book|report>

\chapter 2666 \renewcommand\chapter{%
2667 % \if@openright\cleardoublepage\else\clearpage\fi
2668 % \thispagestyle{plain}%
2669 \CTEX@chapter@break
2670 \thispagestyle{\CTEX@chapter@pagestyle}%
2671 \global\@topnum\z@
2672 % \@afterindentfalse
2673 \ifodd \CTEX@chapter@afterindent
2674   \@afterindenttrue
2675 \else
2676   \@afterindentfalse
2677 \fi
2678 \secdef\@chapter\@schapter}

\@chapter 2679 \def\@chapter[#1]#2{%
2680   \ifnum \c@secnumdepth >\m@ne
2681   <*book>
2682     \if@mainmatter
2683   </book>
2684     \ifodd \CTEX@chapter@numbering
2685       \CTEX@ifnametrue

```

```

2686         \refstepcounter{chapter}%
2687 %         \typeout{\@chapapp\space\thechapter.}%
2688         \typeout{\CTEXthechapter}%
2689 %         \addcontentsline{toc}{chapter}
2690 %         {\protect\numberline{\thechapter}#1}%
2691     \else
2692         \CTEX@ifnamefalse
2693         \CTEX@makeanchor{\Hy@chapapp*}%
2694 %         \addcontentsline{toc}{chapter}{#1}%
2695     \fi
2696 <*book>
2697     \else
2698         \CTEX@ifnamefalse
2699         \CTEX@makeanchor@chapter{\Hy@chapapp*}%
2700 %         \addcontentsline{toc}{chapter}{#1}%
2701     \fi
2702 </book>
2703     \else
2704         \CTEX@ifnamefalse
2705         \CTEX@makeanchor@chapter{\Hy@chapapp*}%
2706 %         \addcontentsline{toc}{chapter}{#1}%
2707     \fi
2708     \CTEX@getttitle{#1}%
2709     \CTEX@addtocline{chapter}{#1}%
2710     \chaptermark{#1}%
2711 % \addtocontents{lof}{\protect\addvspace{10\p@}}%
2712 % \addtocontents{lot}{\protect\addvspace{10\p@}}%
2713     \CTEX@addloflotskip{chapter}%
2714     \if@twocolumn
2715         \topnewpage[\@makechapterhead{#2}]%
2716     \else
2717         \makechapterhead{#2}%
2718     \afterheading
2719     \fi}

\@schapter 2720 \def\@schapter#1{%
2721     \CTEX@ifnamefalse
2722     \CTEX@makeanchor@schapter{\Hy@chapapp*}%
2723     \CTEX@getttitle{#1}%
2724     \if@twocolumn
2725         \topnewpage[\@makeschapterhead{#1}]%
2726     \else
2727         \makeschapterhead{#1}%
2728         \afterheading
2729     \fi}

\@makechapterhead 2730 \def\@makechapterhead#1{%
2731 % \vspace*{50\p@}%
2732     \CTEX@setheadingskip \CTEX@chapter@beforeskip
2733     \ifodd \CTEX@chapter@fixskip \CTEX@fixtopskip \fi
2734     \vspace*{\CTEX@headingskip}%
2735     \begingroup
2736 % \parindent \z@ \raggedright \normalfont
2737     \CTEX@heading@format@initial
2738     \CTEX@chapter@format{%
2739 % \ifnum \c@secnumdepth >\m@ne
2740 % \if@mainmatter
2741 % \huge\bfseries\@chapapp\space\thechapter\par\nobreak\vskip 20\p@
2742 % \fi
2743 % \fi
2744     \CTEX@headinghang{chapter}%
2745     {\CTEX@ifname{\CTEX@chaptername\CTEX@chapter@aftername}{}}%
2746 % \Huge \bfseries #1\par\nobreak
2747     \CTEX@chapter@titleformat{#1}%
2748     \CTEX@chapter@aftertitle}\par
2749 \endgroup
2750 \nobreak

```

```

2751 % \vskip 40\p@
2752 \CTEX@setheadingskip \CTEX@chapter@afterskip
2753 \ifodd \CTEX@chapter@fixskip \CTEX@fixheadingskip \fi
2754 \vskip \CTEX@headingskip}

\@makeschapterhead 2755 \def\@makeschapterhead#1{%
2756 % \vspace*{50\p@}%
2757 \CTEX@setheadingskip \CTEX@chapter@beforeskip
2758 \ifodd \CTEX@chapter@fixskip \CTEX@fixtopskip \fi
2759 \vspace*{\CTEX@headingskip}%
2760 \begingroup
2761 % \parindent \z@ \raggedright \normalfont \interlinepenalty\@M
2762 \CTEX@heading@format@initial
2763 \CTEX@chapter@format{%
2764 \CTEX@headinghang{chapter}}}%
2765 % \Huge \bfseries #1\par\nobreak
2766 \CTEX@chapter@titleformat{#1}%
2767 \CTEX@chapter@aftertitle}\par
2768 \endgroup
2769 \nobreak
2770 % \vskip 40\p@
2771 \CTEX@setheadingskip \CTEX@chapter@afterskip
2772 \ifodd \CTEX@chapter@fixskip \CTEX@fixheadingskip \fi
2773 \vskip \CTEX@headingskip}

2774 </book|report>

```

14.8.2.3 section 类的标题

\@startsection L^AT_EX 的标准参数是：

$\{<\textit{name}>\}\{<\textit{level}>\}\{<\textit{indent}>\}\{<\textit{beforeskip}>\}\{<\textit{afterskip}>\}\{<\textit{style}>\}*[<\textit{altheading}>]\{<\textit{heading}>\}$

```

2775 \def\@startsection#1#2#3#4#5#6{%
2776 \if@noskipsec \leavevmode \fi
2777 \par
2778 % \@tempskipa #4\relax
2779 % \@afterindenttrue
2780 % \ifdim \@tempskipa <\z@
2781 % \@tempskipa -\@tempskipa \@afterindentfalse
2782 % \fi
2783 \CTEX@update@sectionformat@n{#1}%
2784 \ifodd \CTEX@afterindent
2785 \@afterindenttrue
2786 \else
2787 \@afterindentfalse
2788 \fi
2789 \if@nobreak
2790 \everypar{}%
2791 \else
2792 % \addpenalty\@secpenalty\addvspace\@tempskipa
2793 \csname CTEX@#1@break\endcsname
2794 \CTEX@setheadingskip{#4}%
2795 \ifodd \CTEX@fixskip \CTEX@fixheadingskip \fi
2796 \addvspace \CTEX@headingskip
2797 \fi
2798 \@ifstar
2799 {\CTEX@makeanchor@ssect{#1*}\@ssect{#3}{#4}{#5}{#6}}%
2800 {\dblarg{\@sect{#1}{#2}{#3}{#4}{#5}{#6}}}%

\@secntformat 2801 \def\@secntformat#1{%
2802 % \csname the#1\endcsname\quad}%
2803 \csname CTEX@#1name\endcsname
2804 \csname CTEX@#1aftername\endcsname}

\@sect 2805 \def\@sect#1#2#3#4#5#6[#7]#8{%
2806 \ifnum #2>\c@secnumdepth

```



```

2807 \CTEX@ifnamefalse
2808 \CTEX@makeanchor@sect{#1*}%
2809 \let\@svsec\@empty
2810 \else
2811 \ifodd \csname CTEX@#1@numbering\endcsname
2812 \CTEX@ifnametrue
2813 \refstepcounter{#1}%
2814 \protected@edef\@svsec{\@secntformat{#1}\relax}%
2815 \else
2816 \CTEX@ifnamefalse
2817 \CTEX@makeanchor{#1*}%
2818 \let\@svsec\@empty
2819 \fi
2820 \fi
2821 \CTEX@getttitle{#7}%
2822 % \@tempkipa #5\relax
2823 % \ifdim \@tempkipa>\z@
2824 \unless \ifodd \CTEX@runin
2825 \begingroup
2826 \CTEX@heading@format@initial
2827 #6{%
2828 % \@hangfrom{\hskip #3\relax\@svsec}%
2829 % \interlinepenalty \@M #8\@par
2830 \CTEX@sectionhang{#3}{\@svsec}%
2831 \csname CTEX@#1@titleformat\endcsname{#8}%
2832 \csname CTEX@#1@aftertitle\endcsname}\par
2833 \endgroup
2834 \csname #1mark\endcsname{#7}%
2835 % \addcontentsline{toc}{#1}{%
2836 % \ifnum #2>\c@secnumdepth \else
2837 % \protect\numberline{\csname the#1\endcsname}%
2838 % \fi
2839 % #7}%
2840 \CTEX@addtocline{#1}{#7}%
2841 \else
2842 \def\@svsechd{%
2843 #6{%
2844 % \hskip #3\relax \@svsec #8
2845 % {\CTEX@indentbox{#3}}\@svsec
2846 % \csname CTEX@#1@titleformat\endcsname{#8}%
2847 % \csname CTEX@#1@aftertitle\endcsname}%
2848 % \csname #1mark\endcsname{#7}%
2849 % \addcontentsline{toc}{#1}{%
2850 % \ifnum #2>\c@secnumdepth \else
2851 % \protect\numberline{\csname the#1\endcsname}%
2852 % \fi
2853 % #7}%
2854 \CTEX@addtocline{#1}{#7}%
2855 \fi
2856 \@xsect{#5}}

\@ssect 2857 \def\@ssect#1#2#3#4#5{%
2858 \CTEX@ifnamefalse
2859 \CTEX@getttitle{#5}%
2860 % \@tempkipa #3\relax
2861 % \ifdim \@tempkipa>\z@
2862 \unless \ifodd \CTEX@runin
2863 \begingroup
2864 \CTEX@heading@format@initial
2865 #4{%
2866 % \@hangfrom{\hskip #1}%
2867 % \interlinepenalty \@M #5\__ctexpar
2868 \CTEX@sectionhang{#1}{}%
2869 \CTEX@titleformat{n}{#5}%
2870 \CTEX@aftertitle}\par
2871 \endgroup
2872 \else

```

```

2873 % \def\@svsechd{#4{\hskip #1\relax #5}}%
2874 \def\@svsechd{#4{{\CTEX@indentbox{#1}}}%
2875 \CTEX@titleformat@n{#5}\CTEX@aftertitle}}%
2876 \fi
2877 \@xsect{#3}}

\@xsect 2878 \def\@xsect#1{%
2879 % \@tempskipa #1\relax
2880 % \ifdim \@tempskipa>z@
2881 \unless \ifodd \CTEX@runin
2882 \par \nobreak
2883 % \vskip \@tempskipa
2884 \CTEX@setheadingskip{#1}%
2885 \ifodd \CTEX@fixskip \CTEX@fixheadingskip \fi
2886 \vskip \CTEX@headingskip
2887 \@afterheading
2888 \else
2889 \@nobreakfalse
2890 \global\@noskipsectrue
2891 \everypar{%
2892 \if@noskipsec
2893 \global\@noskipsecfalse
2894 {\setbox\z@\lastbox}%
2895 \clubpenalty\@M
2896 \begingroup \@svsechd \endgroup
2897 \unskip
2898 % \@tempskipa #1\relax
2899 % \hskip -\@tempskipa
2900 \CTEX@heading@glue{#1}%
2901 \else
2902 \clubpenalty \@clubpenalty
2903 \everypar{}}%
2904 \fi}%
2905 \fi
2906 \ignorespaces}

```

\CTEX@headinghang 分别用于实现 \part/\chapter 和 \section 类标题的 indent 和 hang 选项。
\CTEX@sectionhang

```

2907 \cs_new_protected:Npn \CTEX@headinghang #1
2908 {
2909 \ctex_heading_hang:cnn
2910 { \CTEX@#1@hang }
2911 { \use:c { \CTEX@#1@indent } }
2912 }
2913 \cs_new_protected:Npn \CTEX@sectionhang
2914 { \ctex_heading_hang:Nnn \CTEX@hang }

```

\ctex_heading_hang:Nnn hang 选项控制是否采用悬挂缩进,同时设置 \parindent。
\ctex_hang_from:n

```

2915 \cs_new_protected:Npn \ctex_heading_hang:Nnn #1#2#3
2916 {
2917 \dim_set:Nn \tex_parindent:D {#2}
2918 \bool_if:NTF #1
2919 { \ctex_hang_from:n }
2920 { \use:n }
2921 { \__ctex_insert_indent: #3 }
2922 }
2923 \cs_new_protected:Npn \ctex_hang_from:n #1
2924 {
2925 \tex_noindent:D
2926 \hbox_set:Nn \l__ctex_tmp_box {#1}
2927 \tex_hangindent:D = \box_wd:N \l__ctex_tmp_box
2928 \box_use_drop:N \l__ctex_tmp_box
2929 }
2930 \cs_generate_variant:Nn \ctex_heading_hang:Nnn { c }

```

\ctex_heading_glue:n
\CTEX@heading@glue

如果缩进 #1 长度为零,就不插入水平间距。

```
2931 \cs_new_protected:Npn \ctex_heading_glue:n #1
2932 {
2933   \group_begin:
2934     \skip_set:Nn \l__ctex_heading_skip {#1}
2935     \dim_compare:nNnF \l__ctex_heading_skip = \c_zero_dim
2936       { \skip_horizontal:N \l__ctex_heading_skip }
2937   \group_end:
2938 }
2939 \cs_new_eq:NN \CTEX@heading@glue \ctex_heading_glue:n
```

\CTEX@update@sectionformat@n

在 \@startsection 中设置 \CTEX@titleformat@n 等为相应函数。

```
2940 \cs_new_protected:Npn \CTEX@update@sectionformat@n #1
2941 {
2942   \cs_set_eq:Nc \CTEX@titleformat@n { CTEX@#1@titleformat }
2943   \cs_set_eq:Nc \CTEX@aftertitle { CTEX@#1@aftertitle }
2944   \cs_set_eq:Nc \CTEX@afterindent { CTEX@#1@afterindent }
2945   \cs_set_eq:Nc \CTEX@fixskip { CTEX@#1@fixskip }
2946   \cs_set_eq:Nc \CTEX@hang { CTEX@#1@hang }
2947   \cs_set_eq:Nc \CTEX@runin { CTEX@#1@runin }
2948 }
2949 \cs_new_eq:NN \CTEX@titleformat@n \use:n
2950 \cs_new_eq:NN \CTEX@aftertitle \prg_do_nothing:
2951 \cs_new_eq:NN \CTEX@afterindent \c_true_bool
2952 \cs_new_eq:NN \CTEX@fixskip \c_false_bool
2953 \cs_new_eq:NN \CTEX@hang \c_true_bool
2954 \cs_new_eq:NN \CTEX@runin \c_false_bool
```

\CTEX@part@tocline
\CTEX@chapter@tocline

```
2955 \cs_new:Npn \CTEX@part@tocline #1#2
2956 {
2957   \CTEXifname
2958     { \CTEXthepart \hspace { 1em } }
2959     { }
2960   #2
2961 }
2962 <*book|report>
2963 \cs_new:Npn \CTEX@chapter@tocline #1#2
2964 {
2965   \CTEXifname
2966     { \protect \numberline { \CTEXthechapter \hspace { .3em } } }
2967     { }
2968   #2
2969 }
2970 </book|report>
```

\CTEXnumberline

```
2971 \cs_new:Npn \CTEXnumberline #1
2972 {
2973   \CTEXifname
2974     { \protect \numberline { \use:c { CTEXthe #1 } } }
2975     { }
2976 }

2977 \int_zero:N \l__ctex_tmp_int
2978 \seq_map_inline:Nn \c__ctex_section_headings_seq
2979 {
2980   \int_incr:N \l__ctex_tmp_int
2981   \cs_gset_protected:cpx {#1}
2982   {
2983     \exp_not:N \@startsection {#1}
2984       { \int_use:N \l__ctex_tmp_int }
2985       { \exp_not:c { CTEX@#1@indent } }
2986       { \exp_not:c { CTEX@#1@beforeskip } }
2987       { \exp_not:c { CTEX@#1@afterskip } }
2988       { \exp_not:N \normalfont \exp_not:c { CTEX@#1@format } }
2989   }
```

```

2990 \cs_new:cpn { CTEX@#1@tocline } ##1##2
2991 { \CTEXnumberline { ##1 } ##2 }
2992 }

```

14.8.2.4 附录标题

```

appendix/name 2993 \ctex_define:n
appendix/number 2994 {
appendix/numbering 2995   appendix .meta:nn = { ctex / appendix } {#1} ,
2996   appendix / name .code:n =
2997   { \ctex_assign_heading_name:nn { appendix } {#1} } ,
2998   appendix / number .tl_set:N = \CTEX@appendix@number ,
2999   appendix / numbering .bool_set:N = \CTEX@appendix@numbering ,
3000   appendix / numbering .initial:n = true
3001 }
3002 \tl_new:N \CTEX@preappendix
3003 \tl_new:N \CTEX@postappendix

\appendix 3004 \cs_new_eq:NN \CTEX@save@appendix \appendix
3005 \cs_gset_protected:Npn \appendix
3006 {
3007   \CTEX@save@appendix
3008   <*article>
3009   \gdef \CTEX@presection { \CTEX@preappendix }
3010   \gdef \CTEX@thesection { \CTEX@appendix@number }
3011   \gdef \CTEX@postsection { \CTEX@postappendix }
3012   \gdef \CTEX@section@numbering { \CTEX@appendix@numbering }
3013   </article>
3014   <*book|report>
3015   \gdef \CTEX@prechapter { \CTEX@preappendix }
3016   \gdef \CTEX@thechapter { \CTEX@appendix@number }
3017   \gdef \CTEX@postchapter { \CTEX@postappendix }
3018   \gdef \CTEX@chapter@numbering { \CTEX@appendix@numbering }
3019   </book|report>
3020 }

```

14.8.2.5 设置 hyperref 宏包的标题锚点

\CTEX@makeanchor 设置超链接跳转锚点,在 hyperref 载入后才有意义。

```

3021 \cs_new_protected:Npn \CTEX@makeanchor #1
3022 { }

```

\c__ctex_headings_cs_seq 保存内部标题命令的 CT_EX 定义,用于随后比较。

```

3023 \seq_const_from_clist:Nn \c__ctex_headings_cs_seq
<article> 3024 { part , spart , sect , ssect }
<book|report> 3025 { part , spart , chapter , schapter , sect , ssect }
3026 \seq_map_inline:Nn \c__ctex_headings_cs_seq
3027 {
3028   \cs_new_eq:cc { CTEX@ #1 } { @ #1 }
3029   \cs_new_eq:cN { CTEX@makeanchor@ #1 } \CTEX@makeanchor
3030 }

```

\CTEX@hyperheadinghook hyperref 会重定义内部标题命令,目的在于为没有编号的标题设置锚点(这一功能受他的 implicit 选项的控制)。我们在上面对标题命令的修改已经包含这一功能,如果这些标题命令在 hyperref 载入之前没有被修改过,则恢复 CT_EX 的定义。

```

3031 \cs_new_protected:Npn \CTEX@hyperheadinghook
3032 {
3033   \group_begin:
3034   \ifHy@implicit
3035     \cs_set_eq:NN \H@old@chapter \Hy@org@chapter
3036     \seq_map_inline:Nn \c__ctex_headings_cs_seq

```

```

3037         {
3038             \cs_if_eq:ccT { H@old@ ##1 } { CTEX@ ##1 }
3039             {
3040                 \cs_gset_eq:cc { @ ##1 } { CTEX@ ##1 }
3041                 \cs_gset_eq:cN { CTEX@makeanchor@ ##1 } \CTEX@makeanchor
3042             }
3043         }
3044     \else:
3045         \seq_map_inline:Nn \c__ctex_headings_cs_seq
3046         { \cs_gset_eq:cN { CTEX@makeanchor@ ##1 } \CTEX@makeanchor }
3047     \fi:
3048 \group_end:
3049 }

3050 \ctex_at_end_package:nn { hyperref }
3051 {
3052     \cs_gset_protected:Npn \CTEX@makeanchor #1
3053     {
3054         \Hy@MakeCurrentHrefAuto {#1}
3055         \Hy@raisedlink
3056         {
3057             \hyper@anchorstart { \@currentHref }
3058             \hyper@anchorend
3059         }
3060     }
3061     \CTEX@hyperheadinghook
3062 }

```

14.8.2.6 兼容 `nameref` 宏包

`\CTEX@getttitle` 在 `nameref` 载入后才有意义,与上述 `hyperref` 的处理类似。

```

3063 \cs_new_protected:Npn \CTEX@getttitle #1
3064 { }
3065 \ctex_at_end_package:nn { nameref }
3066 {
3067     \cs_gset_protected:Npn \CTEX@getttitle { \NR@getttitle }
3068     \seq_map_inline:Nn \c__ctex_headings_cs_seq
3069     {
3070         \cs_if_eq:ccT { NR@ #1 } { CTEX@ #1 }
3071         { \cs_gset_eq:cc { @ #1 } { CTEX@ #1 } }
3072     }
3073 }

```

14.8.2.7 兼容 `titlesec` 宏包

我们修改了 `\@startsection` 的定义,它的第四个(`<beforeskip>`)和第五个(`<afterskip>`)参数的符号不再有特殊意义,改由相应的选项 `afterindent` 和 `runin` 来控制。

引入 `titlesec` 宏包,并且未设置它的 `loadonly` 选项时,`titlesec` 会展开 `section` 类标题获取它们的参数,进行初始设置。我们需要进行一些调整。

`\ctex_titlesec_hook:` `\titleformat` 的设置保存在名为 `\ttlhf@<section>` 的宏中备用,它的内容是

```
\ttlhf@<shape>{\<format>}{\<label>}{\<sep>}{\<before>}{\<after>}
```

我们这里的 `<shape>` 为 `hang` 或者 `runin`。`\titlespacing` 的设置保存在 `\ttls@<section>` 之中,它的内容是

```
{\<left>}{\<right>}{\<before>}{\<after>}{\<afterindent>}
```

其中 `<afterindent>` 为 1 或 0,分别对应是否保留段首缩进。我们需要根据 `CTEX` 的 `runin` 和 `afterindent` 选项调整 `\ttlhf@<shape>` 和 `<afterindent>`。注意,由 `\ttl@extract` 得的

`<before>` 和 `<after>` 的值总是非负的, 而 CTeX 的 `beforeskip` 和 `afterskip` 是可以取负值的, 但我们不打算调整它们了。如果使用了 `titlesec` 的 `indentafter` 等选项, 也不需要调整 `\ttls@<section>`。

```

3074 \cs_new_protected:Npn \ctex_titlesec_hook:
3075 {
3076   \@ifpackagewith { titlesec } { explicit }
3077   {
3078     \cs_set_eq:NN \__ctex_titlesec_format:Nn
3079     \__ctex_titlesec_format_explicit:Nn
3080   }
3081   { }
3082   \clist_map_inline:nn
3083   { indentafter , noindentafter , indentfirst , nonindentfirst }
3084   {
3085     \@ifpackagewith { titlesec } { ##1 }
3086     {
3087       \clist_map_break:n
3088       { \cs_set_eq:NN \__ctex_titlesec_hook:n \__ctex_titlesec_format:n }
3089     }
3090     { }
3091   }
3092   \seq_map_function:NN \c__ctex_section_headings_seq \__ctex_titlesec_hook:n
3093 }
3094 \cs_new_protected:Npn \__ctex_titlesec_hook:n #1
3095 {
3096   \__ctex_titlesec_format:n {#1}
3097   \exp_args:Nc \__ctex_titlesec_spacing:Nn { ttls@#1 } {#1}
3098 }
3099 \cs_new_protected:Npn \__ctex_titlesec_format:n #1
3100 {
3101   \cs_if_free:cF { ttlf@#1 }
3102   { \exp_args:Nc \__ctex_titlesec_format:Nn { ttlf@#1 } {#1} }
3103 }
3104 \cs_new_protected:Npn \__ctex_titlesec_format:Nn #1#2
3105 {
3106   \tl_set:Nx #1
3107   {
3108     \bool_if:cTF { CTEX@#2@runin }
3109     { \exp_not:N \ttlh@runin }
3110     { \exp_not:N \ttlh@hang }
3111     \tl_tail:N #1
3112   }
3113 }
3114 \cs_new_protected:Npn \__ctex_titlesec_format_explicit:Nn #1#2
3115 {
3116   \cs_set_nopar:Npx #1 ##1
3117   {
3118     \bool_if:cTF { CTEX@#2@runin }
3119     { \exp_not:N \ttlh@runin }
3120     { \exp_not:N \ttlh@hang }
3121     \exp_args:No \tl_tail:n { #1 { } }
3122   }
3123 }
3124 \cs_new_protected:Npn \__ctex_titlesec_spacing:Nn #1#2
3125 { \tl_set:Nx #1 { \exp_after:wN \__ctex_titlesec_spacing:nnnnnn #1 {#2} } }
3126 \cs_new:Npn \__ctex_titlesec_spacing:nnnnnn #1#2#3#4#5#6
3127 {
3128   \exp_not:n { {#1} {#2} {#3} {#4} }
3129   { \bool_if:cTF { CTEX@#6@afterindent } { \@ne } { \z@ } }
3130 }
3131 \@ifpackageloaded { titlesec }
3132 { }
3133 {
3134   \ctex_at_end_package:nn { titlesec }
3135   {

```

```

3136     \@ifpackagewith { titlesec } { loadonly }
3137     { }
3138     { \ctex_titlesec_hook: }
3139   }
3140 }

```

让编译时终端显示 \CTEXthechapter, 目录使用 \CTEXtheXXX 编号。

```

3141 \ctex_at_end_package:nn { titlesec }
3142 {
3143   <*book|report>
3144   \tl_set:Nn \ttl@chapterout { \typeout { \CTEXthechapter } }
3145   </book|report>
3146   \cs_if_free:NF \ttl@tocpart
3147   {
3148     \cs_set_protected:Npn \ttl@tocpart
3149     { \tl_set:Nn \ttl@a { \CTEXthepart \hspace { 1em } } }
3150   }
3151   \seq_map_inline:Nn \c__ctex_headings_seq
3152   {
3153     \cs_if_exist:cF { \ttl@toc #1 }
3154     {
3155       \cs_new_protected:cpx { \ttl@toc #1 }
3156       {
3157         \tl_set:Nn \exp_not:N \ttl@a
3158         {
3159           \exp_not:N \protect
3160           \exp_not:N \numberline { \exp_not:c { \CTEXthe #1 } }
3161         }
3162       }
3163     }
3164   }
3165 }

```

14.8.2.8 兼容 titleps 宏包

按照 titleps 宏包的实现机制, \CTEXtheXXX 等宏直到页眉排版时才会被展开, 这可能会造成问题²²。

\ctex_titleps_hook: 我们修改 titleps 包的内部命令 \ttl@settopmark 和 \ttl@setsubmark, 将 \CTEXtheXXX 等加入更新队列中。

```

3166 \group_begin:
3167 \char_set_catcode_other:N \#
3168 \cs_new_protected:Npn \ctex_titleps_hook:
3169 {
3170   \ctex_patch_cmd:Nnn \ttl@settopmark
3171   { \protect \@namedef { the#1 } { \@nameuse { the#1 } } }
3172   {
3173     \protect \@namedef { the#1 } { \@nameuse { the#1 } }
3174     \CTEX@titlepslabel@set {#1}
3175   }
3176   \ctex_patch_cmd:Nnn \ttl@setsubmark
3177   { \protect \@namedef { the#1 } { } }
3178   {
3179     \protect \@namedef { the#1 } { }
3180     \CTEX@titlepslabel@clear {#1}
3181   }
3182   \ctex_patch_cmd:Nnn \ttl@setsubmark
3183   { \protect \@namedef { the#2 } { \@nameuse { the#2 } } }
3184   {
3185     \protect \@namedef { the#2 } { \@nameuse { the#2 } }
3186     \CTEX@titlepslabel@set {#2}
3187   }
3188 }

```

²²<https://github.com/CTeX-org/ctex-kit/issues/217>

```

3187     }
3188   }
3189   \group_end:

```

```

\CTEX@titlepslabel@set
\CTEX@titlepslabel@clear

```

这两个函数要在随后被 `\xdef` 展开来获得 `\CTEXtheXXX` 的内容, 不应该用 `\protected` 来定义。

```

3190 \cs_new:Npn \CTEX@titlepslabel@set #1
3191 {
3192   \cs_if_free:cF { CTEXthe#1 }
3193   { \protect \@namedef { CTEXthe#1 } { \@nameuse { CTEXthe#1 } } }
3194 }
3195 \cs_new:Npn \CTEX@titlepslabel@clear #1
3196 {
3197   \cs_if_free:cF { CTEXthe#1 }
3198   { \protect \@namedef { CTEXthe#1 } { } }
3199 }

```

`titles` 宏包的功能可以由 `titlesec` 的选项 `pagestyles` 引入。

```

3200 \ctex_at_end_package:nn { titlesec }
3201 { \cs_if_free:NF \ttl@settopmark { \ctex_titleps_hook: } }
3202 \ctex_at_end_package:nn { titles } { \ctex_titleps_hook: }

```

除此之外, 也可以使用 `titles` 提供的命令 `\newtitlemark` 来完成:

```

\newtitlemark { \CTEXthechapter }
\newtitlemark { \CTEXthesection }

```

但 `\newtitlemark` 不包含章节间的层次信息, 功能上不及修改内部命令完整。

`\ttl@setifthe` 使 `\iftheXXX` 等命令在页眉设置中可用。

```

3203 \ctex_at_end_package:nn { titles }
3204 {
3205   \cs_set_protected:Npn \ttl@setifthe #1
3206   {
3207     \exp_args:Nco \cs_set:Npn { ifthe #1 }
3208     {
3209       \CTEXifname
3210       { \protect \@firstoftwo }
3211       { \protect \@secondoftwo }
3212     }
3213   }
3214   \seq_map_function:NN \c__ctex_headings_seq \ttl@setifthe
3215 }

```

14.8.3 目录标签的宽度

```

\CTEX@toc@width@n 3216 \cs_new_protected:Npn \CTEX@toc@width@n #1
3217 {
3218   \hbox_set:Nn \l__ctex_tmp_box {#1}
3219   \dim_set:Nn \@tempdima
3220   {
3221     \dim_max:nn { \@tempdima }
3222     { \box_wd:N \l__ctex_tmp_box + \f@size \p@ / 2 }
3223   }
3224 }

```

`\numberline` 为 `\numberline` 命令打补丁, 并兼容 `tocloft` 和 `titletoc` 宏包。

```

\__ctex_patch_toc_width:n

```

这里需要替换 `#` 本身, 因此需要先切换为 `other` 类。表示参数的 `#` 用 `\c_parameter_token` 代替。

```

3225 \group_begin:

```



```

3226 \char_set_catcode_other:N \#
3227 \use:n
3228 {
3229   \group_end:
3230   \ctex_preto_cmd:NnnTF \numberline { \ExplSyntaxOff }
3231   { \CTEX@toc@width@n {#1} }
3232   { }
3233   { \ctex_patch_failure:N \numberline }
3234   \cs_new_protected:Npn \__ctex_patch_toc_width:n \c_parameter_token 1
3235   {
3236     \@ifpackageloaded { \c_parameter_token 1 }
3237     { }
3238     {
3239       \ctex_at_end_package:nn { \c_parameter_token 1 }
3240       {
3241         \ctex_preto_cmd:NnnTF \numberline
3242         { \char_set_catcode_letter:n { 64 } }
3243         { \CTEX@toc@width@n {#1} }
3244         { }
3245         { \ctex_patch_failure:N \numberline }
3246       }
3247     }
3248   }
3249 }
3250 \__ctex_patch_toc_width:n { tocleft }
3251 \__ctex_patch_toc_width:n { titletoc }

```

14.8.4 页眉信息的修改

```

\ps@headings 3252 <*article>
3253 \if@twoside
3254   \ctex_patch_cmd:Nnn \ps@headings
3255   { \ifnum \c@secnumdepth > \z@ \thesection \quad \fi }
3256   { \CTEXifname { \CTEXthesection \quad } { } }
3257   \ctex_patch_cmd:Nnn \ps@headings
3258   { \ifnum \c@secnumdepth > \ne \thesubsection \quad \fi }
3259   { \CTEXifname { \CTEXthesubsection \quad } { } }
3260 \else:

```

不知为何,标准文档类此处对 secnumdepth 的判断为 0,与 \section 的层次 1 不符。

```

3261   \ctex_patch_cmd:Nnn \ps@headings
3262   { \ifnum \c@secnumdepth > \m@ne \thesection \quad \fi }
3263   { \CTEXifname { \CTEXthesection \quad } { } }
3264 \fi:
3265 </article>
3266 <*book|report>
3267 \ctex_patch_cmd:Nnn \ps@headings
3268 {
3269   <book> \ifnum \c@secnumdepth > \m@ne \if@mainmatter
3270   <report> \ifnum \c@secnumdepth > \m@ne
3271     \@chapapp \thechapter . ~ \ %
3272     \fi
3273   <book> \fi \fi
3274 }
3275 { \CTEXifname { \CTEXthechapter \quad } { } }
3276 \if@twoside
3277   \ctex_patch_cmd:Nnn \ps@headings
3278   { \ifnum \c@secnumdepth > \z@ \thesection . ~ \ \fi }
3279   { \CTEXifname { \CTEXthesection \quad } { } }
3280 \fi:
3281 </book|report>

```

\ps@fancy 这里对 fancyhdr 宏包打补丁。原来 fancyhdr 宏包中使用 \thesection 等宏表示页眉中的章节编号,这里改用 ctex 包所用的 \CTEXthesection 系列宏。

```

3282 \ctex_at_end_package:nn { fancyhdr }
3283 {
3284   <*article>
3285     \ctex_patch_cmd:Nnn \ps@fancy
3286       { \ifnum \c@secnumdepth > \z@ \thesection \hskip 1em \relax \fi }
3287       { \CTEXifname { \CTEXthesection \quad } { } }
3288     \ctex_patch_cmd:Nnn \ps@fancy
3289       { \ifnum \c@secnumdepth > \@ne \thesubsection \hskip 1em \relax \fi }
3290       { \CTEXifname { \CTEXthesubsection \quad } { } }
3291   </article>
3292   <*book|report>
3293     \ctex_patch_cmd:Nnn \ps@fancy
3294       { \ifnum \c@secnumdepth > \m@ne \@chapapp \ \thechapter . ~ \ \fi }
3295       { \CTEXifname { \CTEXthechapter \quad } { } }
3296     \ctex_patch_cmd:Nnn \ps@fancy
3297       { \ifnum \c@secnumdepth > \z@ \thesection . ~ \ \fi }
3298       { \CTEXifname { \CTEXthesection \quad } { } }
3299   </book|report>
3300 }

3301 </article|book|report>

```

14.8.5 beamer 标题页模板的修改

```

3302 <*beamer>
3303 \ExplSyntaxOff

    对应 \partpage。

3304 \defbeamertemplate*{part page}{CTEX}[1][]{%
3305   \begingroup
3306   % \centering
3307   % {\usebeamerfont{part name}%
3308   % \usebeamercolor{fg}{part name}\partname~\insertromanpartnumber}
3309   % \vskip1em\par
3310   \par \addvspace{\glueexpr\CTEX@part@beforeskip\relax}%
3311   \CTEX@heading@format@initial
3312   \CTEX@part@format{%
3313     \CTEX@indentbox{\CTEX@part@indent}%
3314     \ifodd \CTEX@part@numbering
3315       \CTEX@partname \CTEX@part@aftername
3316     \fi
3317     \begin{beamercolorbox}[sep=16pt,center,#1]{part title}
3318     % \usebeamerfont{part title}\insertpart\par
3319     \CTEX@part@titleformat \insertpart \CTEX@part@aftertitle
3320     \end{beamercolorbox}}%
3321     \par \addvspace{\glueexpr\CTEX@part@afterskip\relax}%
3322   \endgroup}

```

对应 \sectionpage。

```

3323 \defbeamertemplate*{section page}{CTEX}[1][]{%
3324   \begingroup
3325   % \centering
3326   % {\usebeamerfont{section name}%
3327   % \usebeamercolor{fg}{section name}\sectionname~\insertsectionnumber}
3328   % \vskip1em\par
3329   \par \addvspace{\glueexpr\CTEX@section@beforeskip\relax}%
3330   \CTEX@heading@format@initial
3331   \CTEX@section@format{%
3332     \CTEX@indentbox{\CTEX@section@indent}%
3333     \ifodd \CTEX@section@numbering
3334       \CTEX@sectionname \CTEX@section@aftername
3335     \fi

```

```

3336 \begin{beamercolorbox}[sep=12pt,center,#1]{part title}
3337 % \usebeamerfont{section title}\insertsection\par
3338 \CTEX@section@titleformat \insertsection \CTEX@section@aftertitle
3339 \end{beamercolorbox}}%
3340 \par \addvspace{\glueexpr\CTEX@section@afterskip\relax}%
3341 \endgroup}

```

对应 \subsectionpage。

```

3342 \defbeamerteremplate*{subsection page}{CTEX}[1][ ]{%
3343 \begingroup
3344 % \centering
3345 % {\usebeamerfont{subsection name}%
3346 % \usebeamercolor[fg]{subsection name}\subsectionname~\insertsubsectionnumber}
3347 % \vskip1em\par
3348 \par \addvspace{\glueexpr\CTEX@subsection@beforeskip\relax}%
3349 \CTEX@heading@format@initial
3350 \CTEX@subsection@format{%
3351 \CTEX@indentbox{\CTEX@subsection@indent}%
3352 \ifodd \CTEX@subsection@numbering
3353 \CTEX@subsectionname \CTEX@subsection@aftername
3354 \fi
3355 \begin{beamercolorbox}[sep=8pt,center,#1]{part title}
3356 % \usebeamerfont{subsection title}\insertsubsection\par
3357 \CTEX@subsection@titleformat \insertsubsection \CTEX@subsection@aftertitle
3358 \end{beamercolorbox}}%
3359 \par \addvspace{\glueexpr\CTEX@subsection@afterskip\relax}%
3360 \endgroup}

```

将 beamer 的默认模板重定向为 CTEX 模板。

```

3361 \defbeamerteremplatealias{part page}{default}{CTEX}
3362 \defbeamerteremplatealias{section page}{default}{CTEX}
3363 \defbeamerteremplatealias{subsection page}{default}{CTEX}
3364 \ExplSyntaxOn
3365 </beamer>

```

14.8.6 标签引用数字的汉化

\refstepcounter 对标题进行引用时,设置标签为通过 number 选项设置的形式。

```

3366 \cs_new_protected:Npn \CTEX@setcurrentlabel@n #1
3367 {
3368 \protected@edef \@currentlabel
3369 {
3370 \cs_if_exist:CTF { CTEX@the#1 }
3371 { \exp_args:cc { p@#1 } { CTEX@the#1 } }
3372 { \exp_not:o { \@currentlabel } }
3373 }
3374 }

```

\ctex_varioref_hook: 关于标签引用的宏包可能会修改 \refstepcounter。其中 cleveref 和 hyperref 宏包都会保存之前的定义,并且它们都要求尽可能晚的被载入,所以对我们上述的修改影响不大。需要注意的是 varioref 宏包,如果它在 CTEX 之后被载入,我们之前的修改将会被覆盖。较新版 L^AT_EX 内核已经包含 \labelformat,可以直接使用。

```

3375 \cs_new_protected:Npn \ctex_varioref_hook:
3376 {
3377 \seq_map_inline:Nn \c__ctex_headings_seq
3378 { \ctex_fix_varioref_label:n { ##1 } }
3379 }

```

__ctex_fix_varioref_label:n varioref 宏包的 \labelformat 实际上是定义一个以 \the<#1> 为参数的宏 \p@<#1>。L^AT_EX 在定义计数器 <#1> 时,都会将 \p@<#1> 初始化为 \@empty。如果这个宏非空,说明用户自定义了标签格式,我们就不再修改。这里不能使用 \exp_args:Nnc,因为 c 这种展开格式不会

将参数放在花括号内。而 `\labelformat` 的定义是

```
\def\labelformat#1{\expandafter\def\csname p@#1\endcsname##1}
```

它的第二个参数必须放在花括号内, 否则将会被作为宏的定界符号。

```
3380 \cs_new_protected:Npn \ctex_fix_varioref_label:n #1
3381 {
3382   \tl_if_empty:cT { p@#1 }
3383   { \exp_args:Nnx \labelformat {#1} { \exp_not:c { CTEX@the#1 } } }
3384 }
```

如果 `varioref` 已经被载入, 则使用它来设置。

```
3385 \cs_if_exist:NTF \labelformat
3386 { \ctex_varioref_hook: }
3387 {
3388   \cs_new_eq:NN \CTEX@save@refstepcounter \refstepcounter
3389   \RenewDocumentCommand \refstepcounter { m }
3390   {
3391     \CTEX@save@refstepcounter {#1}
3392     \CTEX@setcurrentlabel@n {#1}
3393   }
3394   \ctex_at_end_package:nn { varioref } { \ctex_varioref_hook: }
3395 }
```

14.8.7 载入 `<scheme>` 文件

```
3396 \ctex_scheme_input:o { \l__ctex_scheme_tl }
3397 </class|heading>
```

14.8.8 标题格式的 `scheme` 定义

下面使用 C_TE_X 文档类的设置方式, `plain` 模拟标准文档类直接定义或以 `\@startsection` 设定的章节标题格式, `chinese` 汉化的标题格式。

```
3398 <*scheme&(article|book|report|beamer)>

3399 \ctex_set:nn { part }
3400 {
3401   aftertitle = \par ,
3402   <*article|book|report>
3403   hang       = false ,
3404   </article|book|report>
3405   <*plain>
3406   name       = \partname \space ,
3407   <*article|book|report>
3408   number     = \thepart ,
3409   </article|book|report>
3410   <*beamer>
3411   number     = \insertromanpartnumber ,
3412   </beamer>
3413   </plain>
3414   <*chinese>
3415   number     = \chinese { part } ,
3416   </chinese>
3417   <*article>
3418   beforekip  = 4ex ,
3419   afterkip   = 3ex ,
3420   </plain>
3421   format     = \raggedright ,
3422   nameformat = \Large \bfseries ,
3423   aftername  = \par \nobreak ,
3424   titleformat = \huge \bfseries ,
3425   afterindent = false
3426   </plain>
```

```

3427 <*chinese>
3428     format      = \Large \bfseries \centering ,
3429     aftername    = \quad ,
3430     afterindent  = true
3431 </chinese>
3432 </article>
3433 <*book|report>
3434     aftername    = \par \vskip 20 \p@ ,
3435     beforekip    = 0pt \@plus 1fil ,
3436     afterskip    = 0pt \@plus 1fil ,
3437     pagestyle    = plain ,
3438     break        = \if@openright \cleardoublepage \else \clearpage \fi ,
3439 <*plain>
3440     format      = \centering ,
3441     nameformat   = \huge \bfseries ,
3442     titleformat  = \Huge \bfseries
3443 </plain>
3444 <*chinese>
3445     format      = \huge \bfseries \centering
3446 </chinese>
3447 </book|report>
3448 <*beamer>
3449     format      = \centering ,
3450     nameformat   = \usebeamerfont { part ~ name }
3451                  \usebeamercolor [fg] { part ~ name } ,
3452     aftername    = \vskip 1em \par ,
3453     titleformat  = \usebeamerfont { part ~ title }
3454 </beamer>
3455 }

3456 <*book|report>
3457 \ctex_set:nn { chapter }
3458 {
3459     pagestyle    = plain ,
3460     aftertitle    = \par ,
3461     hang         = false ,
3462     beforekip    = 50 \p@ ,
3463     afterskip    = 40 \p@ ,
3464     lofskip      = 10 \p@ ,
3465     lotskip      = 10 \p@ ,
3466     break        = \if@openright \cleardoublepage \else \clearpage \fi ,
3467 <*plain>
3468     name         = \chaptername \space ,
3469     number       = \thechapter ,
3470     format       = \raggedright ,
3471     nameformat   = \huge \bfseries ,
3472     aftername    = \par \nobreak \vskip 20 \p@ ,
3473     titleformat  = \Huge \bfseries ,
3474     afterindent  = false ,
3475     tocline      = \CTEXnumberline {#1} #2
3476 </plain>
3477 <*chinese>
3478     number       = \chinese { chapter } ,
3479     format       = \huge \bfseries \centering ,
3480     aftername    = \quad ,
3481     afterindent  = true
3482 </chinese>
3483 }
3484 </book|report>

3485 \ctex_set:nn { section }
3486 {
3487 <*article|book|report>
3488     number       = \thesection ,
3489     aftername    = \quad ,
3490     aftertitle    = \@@par ,
3491     beforekip    = 3.5ex \@plus 1ex \@minus .2ex ,
3492     afterskip    = 2.3ex \@plus .2ex ,

```

```

3493     runin      = false ,
3494     break      = \addpenalty \@secpenalty ,
3495 <*plain>
3496     format      = \Large \bfseries ,
3497     afterindent = false
3498 </plain>
3499 <*chinese>
3500     format      = \Large \bfseries \centering ,
3501     afterindent = true
3502 </chinese>
3503 </article|book|report>
3504 <*beamer>
3505 <*plain>
3506     name        = \sectionname \space ,
3507 </plain>
3508     format      = \centering ,
3509     number      = \insertsectionnumber ,
3510     nameformat   = \usebeamerfont { section ~ name }
3511                  \usebeamercolor [fg] { section ~ name } ,
3512     aftername    = \vskip 1em \par ,
3513     titleformat  = \usebeamerfont { section ~ title } ,
3514     aftertitle   = \par
3515 </beamer>
3516 }

3517 \ctex_set:nn { subsection }
3518 {
3519 <*article|book|report>
3520     number      = \thesubsection ,
3521     format      = \large \bfseries ,
3522     aftername    = \quad ,
3523     aftertitle   = \@@par ,
3524     beforekip    = 3.25ex \@plus 1ex \@minus .2ex ,
3525     afterskip    = 1.5ex \@plus .2ex ,
3526     runin      = false ,
3527     break      = \addpenalty \@secpenalty ,
3528 <*plain>
3529     afterindent = false
3530 </plain>
3531 <*chinese>
3532     afterindent = true
3533 </chinese>
3534 </article|book|report>
3535 <*beamer>
3536 <*plain>
3537     name        = \subsectionname \space ,
3538     number      = \insertsubsectionnumber ,
3539 </plain>
3540 <*chinese>
3541     number      = \arabic { section } . \arabic { subsection } ,
3542 </chinese>
3543     format      = \centering ,
3544     nameformat   = \usebeamerfont { subsection ~ name }
3545                  \usebeamercolor [fg] { subsection ~ name } ,
3546     aftername    = \vskip 1em \par ,
3547     titleformat  = \usebeamerfont { subsection ~ title } ,
3548     aftertitle   = \par
3549 </beamer>
3550 }

3551 <*article|book|report>

3552 \ctex_set:nn { subsubsection }
3553 {
3554     number      = \thesubsubsection ,
3555     format      = \normalsize \bfseries ,
3556     aftername    = \quad ,
3557     aftertitle   = \@@par ,

```

```

3558     beforeskip = 3.25ex \@plus 1ex \@minus .2ex ,
3559     afterskip  = 1.5ex \@plus .2ex ,
3560     runin      = false ,
3561     break      = \addpenalty \@secpenalty ,
3562 <*plain>
3563     afterindent = false
3564 </plain>
3565 <*chinese>
3566     afterindent = true
3567 </chinese>
3568 }

3569 \ctex_set:nn { paragraph }
3570 {
3571     number      = \theparagraph ,
3572     format      = \normalsize \bfseries ,
3573     aftername    = \quad ,
3574     beforeskip  = 3.25ex \@plus 1ex \@minus .2ex ,
3575     break       = \addpenalty \@secpenalty ,
3576 <*plain>
3577     afterindent = false
3578 </plain>
3579 <*chinese>
3580     afterindent = true
3581 </chinese>
3582 }

3583 \ctex_set:nn { subparagraph }
3584 {
3585     number      = \thesubparagraph ,
3586     format      = \normalsize \bfseries ,
3587     aftername    = \quad ,
3588     beforeskip  = 3.25ex \@plus 1ex \@minus .2ex ,
3589     break       = \addpenalty \@secpenalty ,
3590 <*plain>
3591     afterindent = false
3592 </plain>
3593 <*chinese>
3594     afterindent = true
3595 </chinese>
3596 }

```

处理 sub3section 与 sub4section 的格式。

```

3597 \int_compare:nNnTF \g__ctex_section_depth_int > 2
3598 {
3599     \ctex_set:nn { paragraph }
3600     {
3601         aftertitle = \@@par ,
3602         afterskip  = 1ex \@plus .2ex ,
3603         runin      = false
3604     }
3605 }
3606 {
3607     \ctex_set:nn { paragraph }
3608     {
3609         afterskip  = 1em ,
3610         runin      = true
3611     }
3612 }
3613 \int_compare:nNnTF \g__ctex_section_depth_int > 3
3614 {
3615     \ctex_set:nn { subparagraph }
3616     {
3617         aftertitle = \@@par ,
3618         afterskip  = 1ex \@plus .2ex ,
3619         runin      = false
3620     }

```

```

3621 }
3622 {
3623   \ctex_set:nn { subparagraph }
3624   {
3625     afterskip = 1em ,
3626     runin     = true
3627   }
3628 }
3629 \int_compare:nNnTF \g__ctex_section_depth_int > 2
3630 { \ctex_set:nn { subparagraph } { indent = \c_zero_dim } }
3631 { \ctex_set:nn { subparagraph } { indent = \parindent } }

```

处理附录的格式。

```

3632 \ctex_set:nn { appendix }
3633 <*article>
3634 { number = \@Alph \c@section }
3635 </article>
3636 <*book|report>
3637 {
3638   name = \appendixname \space ,
3639   number = \@Alph \c@chapter
3640 }
3641 </book|report>
3642 </article|book|report>
3643 </scheme&(article|book|report|beamer)>

```

14.8.9 ctex.sty 的 heading 选项

```

3644 <*ctex|ctexheading>

```

\c__ctex_std_class_tl 用于记录被引入的标准文档类。

```

3645 \clist_map_inline:nn { article , book , report , beamer }
3646 {
3647   \ifclassloaded {#1}
3648   { \clist_map_break:n { \tl_const:Nn \c__ctex_std_class_tl {#1} } }
3649   { }
3650 }

```

若标准文档类被引入，则载入对应的标题定义文件。否则视 \chapter 是否有定义来引入 book 或者 article。

```

3651 \msg_new:nnn { ctex } { not-standard-class }
3652 {
3653   None~of~the~standard~document~classes~was~loaded.\\
3654   Heading~`#1'~is~selected.\\
3655   ctex~may~not~work~as~expected.
3656 }
3657 <ctex> \bool_if:NTF \l__ctex_heading_bool
3658 <ctexheading> \use:n
3659 {
3660   \tl_if_exist:NTF \c__ctex_std_class_tl
3661   { \cs_new_eq:NN \c__ctex_class_tl \c__ctex_std_class_tl }
3662   {
3663     \cs_if_exist:NTF \chapter
3664     {
3665       \cs_if_exist:NF \if@mainmatter
3666       { \cs_new_eq:NN \if@mainmatter \tex_iftrue:D }
3667       \tl_const:Nn \c__ctex_class_tl { book }
3668     }
3669     { \tl_const:Nn \c__ctex_class_tl { article } }
3670     \msg_warning:nnx { ctex } { not-standard-class } { \c__ctex_class_tl }
3671   }
3672   \ctex_file_input:n { ctex-heading- \c__ctex_class_tl .def }

```



```

3673   }
<ctex> 3674   { \ctex_scheme_input:o { \l__ctex_scheme_tl } }
3675 </ctex|ctexheading>

```

14.8.10 标题配置文件

```

3676 <*name>
3677 \keys_set_known:nn { ctex }
3678 {
3679   contentsname = 目录 ,
3680   listfigurename = 插图 ,
3681   listtablename = 表格 ,
3682   figurename = 图 ,
3683   tablename = 表 ,
3684   abstractname = 摘要 ,
3685   indexname = 索引 ,
3686   bibname = 参考文献 ,
3687   appendixname = 附录 ,
3688   proofname = 证明 ,
3689   algorithmname = 算法 ,
3690   refname = 参考文献 ,
3691   continuation = (续) ,
3692   part / name = { 第 , 部分 } ,
3693   chapter / name = { 第 , 章 }
3694 }
3695 </name>

```

14.9 chinese 方案的其他设置

```

3696 <*scheme&chinese>

```

chinese 在标准文档类下的页面格式总采用 headings。

```

<article|book|report> 3697 \pagestyle { headings }

```

日期格式。

```

3698 \ctex_set:n { today = small }

```

若用户未设置宏包选项 autoindent, 则自动调整首行缩进。

```

3699 \ctex_if_autoindent_touched:F
3700 { \ctex_set:n { autoindent = true } }

```

使用标题定义时的设置。首先是命题名字汉化。beamer 需要汉化定理名称。

```

3701 <!*generic>
3702 \str_if_eq:onTF { \g__ctex_encoding_tl } { GBK }
3703 <*beamer>
3704 {
3705   \uselanguage { ChineseGBK }
3706   \languagealias { chinese } { ChineseGBK }
3707   \ctex_file_input:n { ctex-name-gbk.cfg }
3708 }
3709 {
3710   \uselanguage { ChineseUTF8 }
3711   \languagealias { chinese } { ChineseUTF8 }
3712   \ctex_file_input:n { ctex-name-utf8.cfg }
3713 }

```

让 translator 包优先查找中文翻译。

```

3714 \clist_put_left:Nn \trans@languagepath { chinese }
3715 </beamer>
3716 <!*beamer>
3717 { \ctex_file_input:n { ctex-name-gbk.cfg } }
3718 { \ctex_file_input:n { ctex-name-utf8.cfg } }

```

对 **beamer** 以外的文档类, 若用户未设置宏包选项 `zihao`, 则设置 `\normalsize` 为五号字。**beamer** 不调整默认字体大小。

```
3719 \int_compare:nNnF \g__ctex_font_size_int > { -1 }
3720 { \int_gset:Nn \g__ctex_font_size_int { 0 } }
```

对 **beamer** 以外的文档类, 若用户未设置宏包选项 `linespread`, 则设置行距初始值为 $1.3 \times 1.2 = 1.56$ 倍字体大小。**beamer** 不调整行距。

```
3721 \fp_if_nan:nT { \l__ctex_line_spread_fp }
3722 { \fp_set:Nn \l__ctex_line_spread_fp { 1.3 } }
3723 </!beamer>
3724 </!generic>
```

不使用标题定义时的通用设置, 注意此处 `\c__ctex_std_class_tl` 可能没有定义。

```
3725 <*generic>
3726 \tl_set:Nn \l__ctex_tmp_tl { beamer }
3727 \tl_if_eq:NNTF \c__ctex_std_class_tl \l__ctex_tmp_tl
3728 {
3729   \str_if_eq:onTF { \g__ctex_encoding_tl } { GBK }
3730   {
3731     \uselanguage { ChineseGBK }
3732     \languagealias { chinese } { ChineseGBK }
3733     \ctex_file_input:n { ctex-name-gbk.cfg }
3734   }
3735   {
3736     \uselanguage { ChineseUTF8 }
3737     \languagealias { chinese } { ChineseUTF8 }
3738     \ctex_file_input:n { ctex-name-utf8.cfg }
3739   }
3740   \clist_put_left:Nn \trans@languagepath { chinese }
3741 }
3742 {
3743   \str_if_eq:onTF { \g__ctex_encoding_tl } { GBK }
3744   { \ctex_file_input:n { ctex-name-gbk.cfg } }
3745   { \ctex_file_input:n { ctex-name-utf8.cfg } }
3746   \int_compare:nNnF \g__ctex_font_size_int > { -1 }
3747   { \int_gset:Nn \g__ctex_font_size_int { 0 } }
3748   \fp_if_nan:nT { \l__ctex_line_spread_fp }
3749   { \fp_set:Nn \l__ctex_line_spread_fp { 1.3 } }
```

若 **ctex** 宏包与标准文档类及其衍生文档类联用, 则将载入 `indentfirst` 宏包, 实现章节标题后首个段落的段首缩进。

```
3750 \tl_if_exist:NT \c__ctex_std_class_tl
3751 { \RequirePackage { indentfirst } }
3752 }
3753 </generic>
3754 </scheme&chinese>
```

14.10 中文字号

```
3755 <*class|ctex|ctexsize>
\zihao 3756 \NewDocumentCommand \zihao { m }
3757 { \exp_args:Nx \ctex_zihao:n {#1} \tex_ignorespaces:D }

\ctex_zihao:n 3758 \cs_new_protected:Npn \ctex_zihao:n #1
3759 {
3760   \prop_get:NnNTF \c__ctex_font_size_prop {#1} \l__ctex_font_size_tl
3761   { \exp_after:wN \fontsize \l__ctex_font_size_tl \selectfont }
3762   { \msg_error:nnn { ctex } { fontsize } {#1} }
3763 }
3764 \msg_new:nnnn { ctex } { fontsize }
3765 { Undefined~Chinese~font~size~`#1'~in~command~\token_to_str:N \zihao.}
3766 {
3767   The~old~font~size~is~used~if~you~continue.\\
3768   The~available~font~sizes~are~listed~as~follow.\\
```

```

3769 \seq_use:Nnnn \c__ctex_font_size_seq { ~and~ } { ,~ } { ,~and~ }.
3770 }

```

14.10.1 定义中文字号

基础行距是字号的 1.2 倍,采用 ϵ -TeX 的 `scaling` 运算得到的结果要比简单的 `1.2\dimexpr` 精确²³。

```

3771 \prop_new:N \c__ctex_font_size_prop
3772 \seq_new:N \c__ctex_font_size_seq
3773 \cs_new_protected:Npn \__ctex_save_font_size:nn #1#2
3774 {
3775   \use:x
3776   {
3777     \prop_gput:Nnn \exp_not:N \c__ctex_font_size_prop {#1}
3778     {
3779       { \dim_to_decimal:n {#2} }
3780       { \dim_to_decimal:n { (#2) * 6 / 5 } }
3781     }
3782   }
3783   \seq_gput_right:Nn \c__ctex_font_size_seq {#1}
3784 }
3785 \clist_map_inline:nn
3786 {
3787   { 8 } { 5 bp } ,
3788   { 7 } { 5.5 bp } ,
3789   { -6 } { 6.5 bp } ,
3790   { 6 } { 7.5 bp } ,
3791   { -5 } { 9 bp } ,
3792   { 5 } { 10.5 bp } ,
3793   { -4 } { 12 bp } ,
3794   { 4 } { 14 bp } ,
3795   { -3 } { 15 bp } ,
3796   { 3 } { 16 bp } ,
3797   { -2 } { 18 bp } ,
3798   { 2 } { 22 bp } ,
3799   { -1 } { 24 bp } ,
3800   { 1 } { 26 bp } ,
3801   { -0 } { 36 bp } ,
3802   { 0 } { 42 bp }
3803 }
3804 { \__ctex_save_font_size:nn #1 }

\ctex_declare_math_sizes:nnnn 3805 \cs_new_protected:Npn \ctex_declare_math_sizes:nnnn #1#2#3#4
3806 {
3807   \__ctex_get_font_sizes:Nn \l__ctex_font_size_tl { {#1} {#2} {#3} {#4} }
3808   \exp_after:wN \DeclareMathSizes \l__ctex_font_size_tl
3809 }

\__ctex_get_font_sizes:Nn 3810 \cs_new_protected:Npn \__ctex_get_font_sizes:Nn #1#2
3811 {
3812   \tl_clear:N #1
3813   \tl_map_inline:nn {#2}
3814   {
3815     \prop_get:NnNTF \c__ctex_font_size_prop {##1} \l__ctex_tmp_tl
3816     { \tl_put_right:Nx #1 { { \tl_head:N \l__ctex_tmp_tl } } }
3817     { \tl_put_right:Nx #1 { { \dim_to_decimal:n { ##1 } } } }
3818   }
3819 }

3820 \clist_map_inline:nn
3821 {
3822   { 8 } { 8 } { 5pt } { 5pt } ,
3823   { 7 } { 7 } { 5pt } { 5pt } ,

```

²³<http://thread.gmane.org/gmane.comp.tex.latex.latex3/3190>

```

3824 { -6 }{ -6 }{ 5pt }{ 5pt } ,
3825 { 6 }{ 6 }{ 5pt }{ 5pt } ,
3826 { -5 }{ -5 }{ 6pt }{ 5pt } ,
3827 { 5 }{ 5 }{ 7pt }{ 5pt } ,
3828 { -4 }{ -4 }{ 8pt }{ 6pt } ,
3829 { 4 }{ 4 }{ 5 }{ 6 } ,
3830 { -3 }{ -3 }{ -4 }{ -5 } ,
3831 { 3 }{ 3 }{ 4 }{ 5 } ,
3832 { -2 }{ -2 }{ -3 }{ -4 } ,
3833 { 2 }{ 2 }{ 3 }{ 4 } ,
3834 { -1 }{ -1 }{ -2 }{ -3 } ,
3835 { 1 }{ 1 }{ 2 }{ 3 } ,
3836 { -0 }{ -0 }{ -1 }{ -2 } ,
3837 { 0 }{ 0 }{ 1 }{ 2 }
3838 }
3839 { \ctex_declare_math_sizes:nnnn #1 }

```

14.10.2 修改默认字号大小

```

\ctex_set_font_size:Nnn 3840 \cs_new_protected:Npn \ctex_set_font_size:Nnn #1#2#3
3841 {
3842   \prop_get:NnNTF \c__ctex_font_size_prop {#2} \l__ctex_font_size_tl
3843   { \exp_after:wN \__ctex_set_font_size:nnNn \l__ctex_font_size_tl #1 {#3} }
3844   { \msg_error:nnn { ctex } { fontsize } {#2} }
3845 }
3846 \cs_new_protected:Npn \__ctex_set_font_size:nnNn #1#2#3#4
3847 { \cs_set_protected:Npn #3 { \@setfontsize #3 {#1} {#2} #4 } }

3848 \int_case:nn { \g__ctex_font_size_int }
3849 {
3850   { 0 } { \ctex_file_input:n { ctex-c5size.clo } }
3851   { 1 } { \ctex_file_input:n { ctex-cs4size.clo } }
3852 }

3853 </class>ctex<ctexsize>

3854 <*c5size>
3855 \ctex_set_font_size:Nnn \normalsize { 5 }
3856 {
3857   \abovedisplayskip 10\p@ \@plus2\p@ \@minus5\p@
3858   \abovedisplayshortskip \z@ \@plus3\p@
3859   \belowdisplayshortskip 6\p@ \@plus3\p@ \@minus3\p@
3860   \belowdisplayskip \abovedisplayskip
3861   \let\@listi\@listI
3862 }
3863 \ctex_set_font_size:Nnn \small { -5 }
3864 {
3865   \abovedisplayskip 8.5\p@ \@plus3\p@ \@minus4\p@
3866   \abovedisplayshortskip \z@ \@plus2\p@
3867   \belowdisplayshortskip 4\p@ \@plus2\p@ \@minus2\p@
3868   \def\@listi{\leftmargin\leftmarginI
3869             \topsep 4\p@ \@plus2\p@ \@minus2\p@
3870             \parsep 2\p@ \@plus\p@ \@minus\p@
3871             \itemsep \parsep}
3872   \belowdisplayskip \abovedisplayskip
3873 }
3874 \ctex_set_font_size:Nnn \footnotesize { 6 }
3875 {
3876   \abovedisplayskip 6\p@ \@plus2\p@ \@minus4\p@
3877   \abovedisplayshortskip \z@ \@plus\p@
3878   \belowdisplayshortskip 3\p@ \@plus\p@ \@minus2\p@
3879   \def\@listi{\leftmargin\leftmarginI
3880             \topsep 3\p@ \@plus\p@ \@minus\p@
3881             \parsep 2\p@ \@plus\p@ \@minus\p@
3882             \itemsep \parsep}
3883   \belowdisplayskip \abovedisplayskip
3884 }

```

```

3885 \ctex_set_font_size:Nnn \scriptsize { -6 } { }
3886 \ctex_set_font_size:Nnn \tiny { 7 } { }
3887 \ctex_set_font_size:Nnn \large { -4 } { }
3888 \ctex_set_font_size:Nnn \Large { -3 } { }
3889 \ctex_set_font_size:Nnn \LARGE { -2 } { }
3890 \ctex_set_font_size:Nnn \huge { 2 } { }
3891 \ctex_set_font_size:Nnn \Huge { 1 } { }
3892 </cs5size>
3893 <*cs4size>
3894 \ctex_set_font_size:Nnn \normalsize { -4 }
3895 {
3896   \abovedisplayskip 12\p@ \@plus3\p@ \@minus7\p@
3897   \abovedisplayshortskip \z@ \@plus3\p@
3898   \belowdisplayshortskip 6.5\p@ \@plus3.5\p@ \@minus3\p@
3899   \belowdisplayskip \abovedisplayskip
3900   \let\@listi\@listI
3901 }
3902 \ctex_set_font_size:Nnn \small { 5 }
3903 {
3904   \abovedisplayskip 11\p@ \@plus3\p@ \@minus6\p@
3905   \abovedisplayshortskip \z@ \@plus3\p@
3906   \belowdisplayshortskip 6.5\p@ \@plus3.5\p@ \@minus3\p@
3907   \def\@listi{\leftmargin\leftmargini
3908             \topsep 9\p@ \@plus3\p@ \@minus5\p@
3909             \parsep 4.5\p@ \@plus2\p@ \@minus\p@
3910             \itemsep \parsep}
3911   \belowdisplayskip \abovedisplayskip
3912 }
3913 \ctex_set_font_size:Nnn \footnotesize { -5 }
3914 {
3915   \abovedisplayskip 10\p@ \@plus2\p@ \@minus5\p@
3916   \abovedisplayshortskip \z@ \@plus3\p@
3917   \belowdisplayshortskip 6\p@ \@plus3\p@ \@minus3\p@
3918   \def\@listi{\leftmargin\leftmargini
3919             \topsep 6\p@ \@plus2\p@ \@minus2\p@
3920             \parsep 3\p@ \@plus2\p@ \@minus\p@
3921             \itemsep \parsep}
3922   \belowdisplayskip \abovedisplayskip
3923 }
3924 \ctex_set_font_size:Nnn \scriptsize { 6 } { }
3925 \ctex_set_font_size:Nnn \tiny { -6 } { }
3926 \ctex_set_font_size:Nnn \large { -3 } { }
3927 \ctex_set_font_size:Nnn \Large { -2 } { }
3928 \ctex_set_font_size:Nnn \LARGE { 2 } { }
3929 \ctex_set_font_size:Nnn \huge { -1 } { }
3930 \ctex_set_font_size:Nnn \Huge { 1 } { }
3931 </cs4size>
<ctexsize> 3932 \normalsize
3933 <*classctex>

```

14.11 更新行距

`\l__ctex_line_spread_fp` 被设置了才有必要更新行距和 `\footnotesep`。

```

3934 \fp_if_nan:nF { \l__ctex_line_spread_fp }
3935 {
3936   \exp_args:Nx \linespread { \fp_use:N \l__ctex_line_spread_fp }

```

`\footnotesep` 我们调整了行距，可能导致脚注的间距与行距不协调，需要调整 `\footnotesep`。标准文档类对 `\footnotesep` 的设置是，字体大小为 `\footnotesize` 时 `\strutbox` 的高度（默认值是 `.7\baselineskip`）。我们沿用这个设置方法，只需要更新具体的大小。

```

3937   \group_begin: \footnotesize \exp_args:NNNo \group_end:
3938   \dim_set:Nn \footnotesep { \dim_use:N \box_ht:N \strutbox }
3939 }

```

激活默认字体大小,更新行距、`\parindent` 和 `\CJKglue`。

```
3940 \normalsize
```

14.12 其他功能

`\CTeX` `ctex-faq.sty` 中的定义是

```
\DeclareRobustCommand\CTeX{$\mathbb{C}$\kern-.05em\TeX}
```

然而 `\mathbb` 未必有定义,这里就不采用它了,只定义最简单的形式。`CTeX` 可以直接用在 PDF 书签中。

```
3941 \NewDocumentCommand \CTeX { }
3942 { C \TeX }
3943 \ctex_at_end_package:nn { hyperref }
3944 { \pdfstringdefDisableCommands { \tl_set:Nn \CTeX { CTeX } } }
```

`captiondelimiter` 过时选项。

```
3945 \ctex_define:n
3946 {
3947     captiondelimiter .code:n =
3948     {
3949         \ctex_deprecated_option:n
3950         { You~can~load~the~package~`caption'~to~get~its~functionality. }
3951     }
3952 }
3953 </class|ctex>
```

14.12.1 列表环境的缩进

`\verse`
`\quotation` 只在使用文档类的时候修改诗歌和引用环境的缩进。

```
3954 <*scheme&chinese&(article|book|report)>
3955 \ctex_patch_cmd:Nnn \verse { -1.5em } { -2 \ccwd }
3956 \ctex_patch_cmd:Nnn \verse { 1.5em } { 2 \ccwd }
3957 \ctex_patch_cmd:Nnn \quotation { 1.5em } { 2 \ccwd }
3958 </scheme&chinese&(article|book|report)>
3959 <*class|ctex>
```

14.13 载入中文字库

`\ctex_fontset_error:n` 字库不可用时给出紧急错误信息,停止读取定义文件。

```
3960 \cs_new_protected:Npn \ctex_fontset_error:n #1
3961 { \msg_critical:nnn { ctex } { fontset-unavailable } {#1} }
3962 \msg_new:nnn { ctex } { fontset-unavailable }
3963 { CTeX~fontset~`#1'~is~unavailable~in~current~mode. }
```

`\ctex_fontset_case:nnn` 3 个参数依次为 `pdfTeX`、`upTeX` 和 `XYTeX/LuaTeX`。

```
3964 \cs_new:Npx \ctex_fontset_case:nnn #1#2#3
3965 {
3966     \sys_if_engine_pdftex:TF
3967     {#1}
3968     { \sys_if_engine_uptex:TF {#2} {#3} }
3969 }
```

`\ctex_fontset_case:nnnn` 4 个参数依次为 pdfTeX(生成 PDF)、pdfTeX(生成 DVI)、upTeX 和 XeTeX/LuaTeX。

```
3970 \cs_new:Npx \ctex_fontset_case:nnnn #1#2#3#4
3971 {
3972   \sys_if_engine_pdftex:TF
3973     { \sys_if_output_pdf:TF {#1} {#2} }
3974     { \sys_if_engine_uptex:TF {#3} {#4} }
3975 }
```

`\ctex_detect_platform:` 根据操作系统判断默认字体配置。

```
3976 \cs_new_protected:Npn \ctex_detect_platform:
3977 {
3978   \sys_if_platform_windows:TF
3979     { \tl_gset:Nn \g__ctex_fontset_tl { windows } }
3980     {
3981       \ctex_if_platform_macos:TF
3982         { \tl_gset:Nn \g__ctex_fontset_tl { mac } }
3983         { \tl_gset:Nn \g__ctex_fontset_tl { fandol } }
3984     }
3985 }
```

`\ctex_if_platform_macos:TF` 以特定字体判断 macOS 系统。

```
3986 \cs_new_protected:Npn \ctex_if_platform_macos:TF
3987 { \file_if_exist:nTF { /System/Library/Fonts/Menlo.ttc } }
```

`\ctex_load_fontset:` 如果用户没有指定字体,则探测操作系统,载入相应的字体配置。

```
3988 \cs_new_protected:Npn \ctex_load_fontset:
3989 {
3990   \tl_if_empty:NTF \g__ctex_fontset_tl
3991     { \ctex_detect_platform: }
3992     {
3993       \bool_lazy_or:nnTF
3994         { \str_if_eq_p:on { \g__ctex_fontset_tl } { windowsnew } }
3995         { \str_if_eq_p:on { \g__ctex_fontset_tl } { windowsold } }
3996         {
3997           \msg_warning:nnxx { ctex } { deprecated-fontset }
3998           { \g__ctex_fontset_tl } { windows }
3999           \tl_gset:Nn \g__ctex_fontset_tl { windows }
4000         }
4001       {
4002         \file_if_exist:nF { ctex-fontset- \g__ctex_fontset_tl .def }
4003         {
4004           \use:x
4005           {
4006             \ctex_detect_platform:
4007             \msg_error:nnxx { ctex } { fontset-not-found }
4008             { \g__ctex_fontset_tl } { \exp_not:N \g__ctex_fontset_tl }
4009           }
4010         }
4011       }
4012     }
4013   \ctex_file_input:n { ctex-fontset- \g__ctex_fontset_tl .def }
4014 }
4015 \msg_new:nnn { ctex } { deprecated-fontset }
4016 { CTeX~fontset~`#1'~is~deprecated.\\ Fontset~`#2'~will~be~used~instead. }
4017 \msg_new:nnnn { ctex } { fontset-not-found }
4018 { CTeX~fontset~`#1'~could~not~be~found.\\ Fontset~`#2'~will~be~used~instead. }
4019 { You~may~run~`mktexlsr'~firstly. }
4020 \@onlypreamble \ctex_load_fontset:
```

fontset 在导言区通过 `\ctexset` 载入中文字库的选项。

```
4021 \ctex_define:n
4022 {
```

```

4023 fontset .code:n =
4024 {
4025   \ctex_if_preamble:TF
4026   {
4027     \str_if_eq:eeTF {#1} { none }
4028     { \msg_warning:nnn { ctex } { invalid-value } {#1} }
4029     {
4030       \str_if_eq:onTF { \g__ctex_fontset_tl } { none }
4031       {
4032         \tl_gset:Nx \g__ctex_fontset_tl {#1}
4033         \ctex_load_fontset:
4034       }
4035       {
4036         \msg_error:nnxx { ctex } { fontset-loaded }
4037         { \g__ctex_fontset_tl } {#1}
4038       }
4039     }
4040   }
4041   { \msg_error:nn { ctex } { fontset-only-preamble } }
4042 }
4043 }
4044 \msg_new:nnnn { ctex } { fontset-loaded }
4045 {
4046   CTeX~fontset~`#1'~has~been~loaded.
4047   \str_if_eq:nnF {#1} {#2} { \ Fontset~`#2'~will~be~ignored. }
4048 }
4049 { Only~one~fontset~can~be~loaded~in~the~preamble. }
4050 \msg_new:nnn { ctex } { fontset-only-preamble }
4051 { The~fontset'option~can~be~used~only~in~preamble. }

```

载入中文字库。

```

4052 \str_if_eq:onF { \g__ctex_fontset_tl } { none }
4053 { \ctex_load_fontset: }

```

14.14 宏包配置文件

14.14.1 ctex.cfg

```

4054 \ctex_at_end:n { \ctex_file_input:n { ctex.cfg } }
4055 </class>ctex>
4056 <*config>
4057 %%
4058 </config>

```

14.14.2 ctexopts.cfg

这里仅为配置文件示例：使用 Windows 下的字体设置。

```

4059 <*ctexopts>
4060 %%
4061 %% \ctex_set:nn { option } { fontset = windows }
4062 </ctexopts>

```

14.15 字体定义文件

14.15.1 传统定义方式

```

4063 <*c19[c70]>
4064 %%
4065 %% Chinese characters
4066 %%
<c19> 4067 %% character set: GBK (extension of GB 2312)

```



```

<c70> 4068 %% character set: Unicode
      4069 %% font encoding: Unicode
      4070 %%
      4071 </c19><c70>

```

CJK 宏包使用的字体族。

```

<rm&c19> 4072 \DeclareFontFamily{C19}{rm}{\hyphenchar\font\m@ne}
<rm&c70> 4073 \DeclareFontFamily{C70}{rm}{\hyphenchar\font\m@ne}
<sf&c19> 4074 \DeclareFontFamily{C19}{sf}{\hyphenchar\font\m@ne}
<sf&c70> 4075 \DeclareFontFamily{C70}{sf}{\hyphenchar\font\m@ne}
<tt&c19> 4076 \DeclareFontFamily{C19}{tt}{\hyphenchar\font\m@ne}
<tt&c70> 4077 \DeclareFontFamily{C70}{tt}{\hyphenchar\font\m@ne}

```

up \LaTeX 使用的字体族。up \LaTeX 在 NFSS 下使用字体编码 JY2 和 JT2 来分别表示横排与直排的日文。

```

<rm&jy2> 4078 \DeclareKanjiFamily{JY2}{zhrm}{\}
<rm&jt2> 4079 \DeclareKanjiFamily{JT2}{zhrm}{\}
<sf&jy2> 4080 \DeclareKanjiFamily{JY2}{zhstf}{\}
<sf&jt2> 4081 \DeclareKanjiFamily{JT2}{zhstf}{\}
<tt&jy2> 4082 \DeclareKanjiFamily{JY2}{zhstt}{\}
<tt&jt2> 4083 \DeclareKanjiFamily{JT2}{zhstt}{\}

4084 <*rm>
4085 <*c19>
4086 \DeclareFontShape{C19}{rm}{m}{n}{<-> CJK * gbksong}{\CJKnormal}
4087 \DeclareFontShape{C19}{rm}{b}{n}{<-> CJK * gbkhei}{\CJKnormal}
4088 \DeclareFontShape{C19}{rm}{bx}{n}{<-> CJK * gbkhei}{\CJKnormal}
4089 \DeclareFontShape{C19}{rm}{m}{sl}{<-> CJK * gbksongsl}{\CJKnormal}
4090 \DeclareFontShape{C19}{rm}{b}{sl}{<-> CJK * gbkheisl}{\CJKnormal}
4091 \DeclareFontShape{C19}{rm}{bx}{sl}{<-> CJK * gbkheisl}{\CJKnormal}
4092 \DeclareFontShape{C19}{rm}{m}{it}{<-> CJK * gbkkai}{\CJKnormal}
4093 \DeclareFontShape{C19}{rm}{b}{it}{<-> CJKb * gbkkai}{\CJKbold}
4094 \DeclareFontShape{C19}{rm}{bx}{it}{<-> CJKb * gbkkai}{\CJKbold}
4095 </c19>
4096 <*c70>
4097 \DeclareFontShape{C70}{rm}{m}{n}{<-> CJK * unisong}{\CJKnormal}
4098 \DeclareFontShape{C70}{rm}{b}{n}{<-> CJK * unihei}{\CJKnormal}
4099 \DeclareFontShape{C70}{rm}{bx}{n}{<-> CJK * unihei}{\CJKnormal}
4100 \DeclareFontShape{C70}{rm}{m}{sl}{<-> CJK * unisongsl}{\CJKnormal}
4101 \DeclareFontShape{C70}{rm}{b}{sl}{<-> CJK * uniheisl}{\CJKnormal}
4102 \DeclareFontShape{C70}{rm}{bx}{sl}{<-> CJK * uniheisl}{\CJKnormal}
4103 \DeclareFontShape{C70}{rm}{m}{it}{<-> CJK * unikai}{\CJKnormal}
4104 \DeclareFontShape{C70}{rm}{b}{it}{<-> CJKb * unikai}{\CJKbold}
4105 \DeclareFontShape{C70}{rm}{bx}{it}{<-> CJKb * unikai}{\CJKbold}
4106 </c70>
4107 <*jy2>
4108 \DeclareFontShape{JY2}{zhrm}{m}{n}{<-> upzhserif-h}{\}
4109 \DeclareFontShape{JY2}{zhrm}{m}{it}{<-> upzhserifit-h}{\}
4110 \DeclareFontShape{JY2}{zhrm}{b}{n}{<-> upzhserifb-h}{\}
4111 \DeclareFontShape{JY2}{zhrm}{bx}{n}{<-> upzhserifb-h}{\}
4112 </jy2>
4113 <*jt2>
4114 \DeclareFontShape{JT2}{zhrm}{m}{n}{<-> upzhserif-v}{\}
4115 \DeclareFontShape{JT2}{zhrm}{m}{it}{<-> upzhserifit-v}{\}
4116 \DeclareFontShape{JT2}{zhrm}{b}{n}{<-> upzhserifb-v}{\}
4117 \DeclareFontShape{JT2}{zhrm}{bx}{n}{<-> upzhserifb-v}{\}
4118 </jt2>
4119 </rm>

4120 <*sf>
4121 <*c19>
4122 \DeclareFontShape{C19}{sf}{m}{n}{<-> CJK * gbkyou}{\CJKnormal}
4123 \DeclareFontShape{C19}{sf}{b}{n}{<-> CJKb * gbkyou}{\CJKbold}
4124 \DeclareFontShape{C19}{sf}{bx}{n}{<-> CJKb * gbkyou}{\CJKbold}
4125 \DeclareFontShape{C19}{sf}{m}{sl}{<-> CJK * gbkyousl}{\CJKnormal}
4126 \DeclareFontShape{C19}{sf}{b}{sl}{<-> CJKb * gbkyousl}{\CJKbold}
4127 \DeclareFontShape{C19}{sf}{bx}{sl}{<-> CJKb * gbkyousl}{\CJKbold}

```

```

4128 \DeclareFontShape{C19}{sf}{m}{it}{<-> CJK * gbkyou}{\CJKnormal}
4129 \DeclareFontShape{C19}{sf}{b}{it}{<-> CJKb * gbkyou}{\CJKbold}
4130 \DeclareFontShape{C19}{sf}{bx}{it}{<-> CJKb * gbkyou}{\CJKbold}
4131 </c19>
4132 <*c70>
4133 \DeclareFontShape{C70}{sf}{m}{n}{<-> CJK * uniyou}{\CJKnormal}
4134 \DeclareFontShape{C70}{sf}{b}{n}{<-> CJKb * uniyou}{\CJKbold}
4135 \DeclareFontShape{C70}{sf}{bx}{n}{<-> CJKb * uniyou}{\CJKbold}
4136 \DeclareFontShape{C70}{sf}{m}{sl}{<-> CJK * uniyousl}{\CJKnormal}
4137 \DeclareFontShape{C70}{sf}{b}{sl}{<-> CJKb * uniyousl}{\CJKbold}
4138 \DeclareFontShape{C70}{sf}{bx}{sl}{<-> CJKb * uniyousl}{\CJKbold}
4139 \DeclareFontShape{C70}{sf}{m}{it}{<-> CJK * uniyou}{\CJKnormal}
4140 \DeclareFontShape{C70}{sf}{b}{it}{<-> CJKb * uniyou}{\CJKbold}
4141 \DeclareFontShape{C70}{sf}{bx}{it}{<-> CJKb * uniyou}{\CJKbold}
4142 </c70>
4143 <*jy2>
4144 \DeclareFontShape{JY2}{zhfsf}{m}{n}{<-> upzhsans-h}{\}
4145 \DeclareFontShape{JY2}{zhfsf}{b}{n}{<-> upzhsansb-h}{\}
4146 \DeclareFontShape{JY2}{zhfsf}{bx}{n}{<-> upzhsansb-h}{\}
4147 </jy2>
4148 <*jt2>
4149 \DeclareFontShape{JT2}{zhfsf}{m}{n}{<-> upzhsans-v}{\}
4150 \DeclareFontShape{JT2}{zhfsf}{b}{n}{<-> upzhsansb-v}{\}
4151 \DeclareFontShape{JT2}{zhfsf}{bx}{n}{<-> upzhsansb-v}{\}
4152 </jt2>
4153 </sf>
4154 <*tt>
4155 <*c19>
4156 \DeclareFontShape{C19}{tt}{m}{n}{<-> CJK * gbkfs}{\CJKnormal}
4157 \DeclareFontShape{C19}{tt}{b}{n}{<-> CJKb * gbkfs}{\CJKbold}
4158 \DeclareFontShape{C19}{tt}{bx}{n}{<-> CJKb * gbkfs}{\CJKbold}
4159 \DeclareFontShape{C19}{tt}{m}{sl}{<-> CJK * gbkfssl}{\CJKnormal}
4160 \DeclareFontShape{C19}{tt}{b}{sl}{<-> CJKb * gbkfssl}{\CJKbold}
4161 \DeclareFontShape{C19}{tt}{bx}{sl}{<-> CJKb * gbkfssl}{\CJKbold}
4162 \DeclareFontShape{C19}{tt}{m}{it}{<-> CJK * gbkfs}{\CJKnormal}
4163 \DeclareFontShape{C19}{tt}{b}{it}{<-> CJKb * gbkfs}{\CJKbold}
4164 \DeclareFontShape{C19}{tt}{bx}{it}{<-> CJKb * gbkfs}{\CJKbold}
4165 </c19>
4166 <*c70>
4167 \DeclareFontShape{C70}{tt}{m}{n}{<-> CJK * unifs}{\CJKnormal}
4168 \DeclareFontShape{C70}{tt}{b}{n}{<-> CJKb * unifs}{\CJKbold}
4169 \DeclareFontShape{C70}{tt}{bx}{n}{<-> CJKb * unifs}{\CJKbold}
4170 \DeclareFontShape{C70}{tt}{m}{sl}{<-> CJK * unifssl}{\CJKnormal}
4171 \DeclareFontShape{C70}{tt}{b}{sl}{<-> CJKb * unifssl}{\CJKbold}
4172 \DeclareFontShape{C70}{tt}{bx}{sl}{<-> CJKb * unifssl}{\CJKbold}
4173 \DeclareFontShape{C70}{tt}{m}{it}{<-> CJK * unifs}{\CJKnormal}
4174 \DeclareFontShape{C70}{tt}{b}{it}{<-> CJKb * unifs}{\CJKbold}
4175 \DeclareFontShape{C70}{tt}{bx}{it}{<-> CJKb * unifs}{\CJKbold}
4176 </c70>
4177 <*jy2>
4178 \DeclareFontShape{JY2}{zhhtt}{m}{n}{<-> upzhmono-h}{\}
4179 </jy2>
4180 <*jt2>
4181 \DeclareFontShape{JT2}{zhhtt}{m}{n}{<-> upzhmono-v}{\}
4182 </jt2>
4183 </tt>

```

14.15.2 预定义字库

```
4184 <*fontset>
```

14.15.2.1 adobe

\pdfmapline 不支持 OpenType 字体，因而 adobe 字体集在 pdf 模式下就没有定义。fandol 的情况类似。

```
4185 <*adobe>
```

```

4186 \ctex_fontset_case:nnnn
4187 { \ctex_fontset_error:n { adobe } }
4188 {
4189   \ctex_zhmap_case:nnn
4190   {
4191     \setCJKmainfont { AdobeSongStd-Light.otf }
4192     [
4193       cmap          = UniGB-UTF16-H,
4194       BoldFont      = AdobeHeitiStd-Regular.otf,
4195       ItalicFont    = AdobeKaitiStd-Regular.otf
4196     ]
4197     \setCJKsansfont { AdobeHeitiStd-Regular.otf }
4198     [ cmap = UniGB-UTF16-H ]
4199     \setCJKmonofont { AdobeFangsongStd-Regular.otf }
4200     [ cmap = UniGB-UTF16-H ]
4201     \setCJKfamilyfont { zhsong } { AdobeSongStd-Light.otf }
4202     [ cmap = UniGB-UTF16-H ]
4203     \setCJKfamilyfont { zhhei } { AdobeHeitiStd-Regular.otf }
4204     [ cmap = UniGB-UTF16-H ]
4205     \setCJKfamilyfont { zhkai } { AdobeKaitiStd-Regular.otf }
4206     [ cmap = UniGB-UTF16-H ]
4207     \setCJKfamilyfont { zhfs } { AdobeFangsongStd-Regular.otf }
4208     [ cmap = UniGB-UTF16-H ]
4209     \ctex_punct_set:n { adobe }
4210     \ctex_punct_map_family:nn { \CJKrmdefault } { zhsong }
4211     \ctex_punct_map_family:nn { \CJKsfdefault } { zhhei }
4212     \ctex_punct_map_family:nn { \CJKttdefault } { zhfs }
4213     \ctex_punct_map_bfseries:nn { \CJKrmdefault } { zhhei }
4214     \ctex_punct_map_itshape:nn { \CJKrmdefault } { zhkai }
4215   }
4216   {
4217     \ctex_load_zhmap:nnnn { rm } { zhhei } { zhfs } { zhadobefonts }
4218     \ctex_punct_set:n { adobe }
4219     \ctex_punct_map_family:nn { \CJKrmdefault } { zhsong }
4220     \ctex_punct_map_bfseries:nn { \CJKrmdefault } { zhhei }
4221     \ctex_punct_map_itshape:nn { \CJKrmdefault } { zhkai }
4222   }
4223   { \ctex_fontset_error:n { adobe } }
4224 }
4225 {
4226   \ctex_set_upfonts:nnnnnn
4227   { AdobeSongStd-Light.otf }
4228   { AdobeHeitiStd-Regular.otf }
4229   { AdobeKaitiStd-Regular.otf }
4230   { AdobeHeitiStd-Regular.otf }
4231   { AdobeHeitiStd-Regular.otf }
4232   { AdobeFangsongStd-Regular.otf }
4233   \ctex_set_upfamily:nnn { zhsong } { upzhserif } {}
4234   \ctex_set_upfamily:nnn { zhhei } { upzhsans } {}
4235   \ctex_set_upfamily:nnn { zhfs } { upzhmono } {}
4236   \ctex_set_upfamily:nnn { zhkai } { upzhserifit } {}
4237 }
4238 {
4239   \setCJKmainfont { AdobeSongStd-Light }
4240   [ BoldFont = AdobeHeitiStd-Regular, ItalicFont = AdobeKaitiStd-Regular ]
4241   \setCJKsansfont { AdobeHeitiStd-Regular }
4242   \setCJKmonofont { AdobeFangsongStd-Regular }
4243   \setCJKfamilyfont { zhsong } { AdobeSongStd-Light }
4244   \setCJKfamilyfont { zhhei } { AdobeHeitiStd-Regular }
4245   \setCJKfamilyfont { zhfs } { AdobeFangsongStd-Regular }
4246   \setCJKfamilyfont { zhkai } { AdobeKaitiStd-Regular }
4247 }
4248 </adobe>

```

14.15.2.2 fandol

```

4249 <fandol>
4250 \ctex_fontset_case:nnnn
4251 { \ctex_fontset_error:n { fandol } }
4252 {
4253   \ctex_zhmap_case:nnn
4254   {
4255     \setCJKmainfont { FandolSong-Regular.otf }
4256     [
4257       cmap          = UniGB-UTF16-H,
4258       BoldFont      = FandolSong-Bold.otf,
4259       ItalicFont    = FandolKai-Regular.otf
4260     ]
4261     \setCJKsansfont { FandolHei-Regular.otf }
4262     [ cmap = UniGB-UTF16-H, BoldFont = FandolHei-Bold.otf ]
4263     \setCJKmonofont { FandolFang-Regular.otf }
4264     [ cmap = UniGB-UTF16-H ]
4265     \setCJKfamilyfont { zhsong } { FandolSong-Regular.otf }
4266     [ cmap = UniGB-UTF16-H, BoldFont = FandolSong-Bold.otf ]
4267     \setCJKfamilyfont { zhhei } { FandolHei-Regular.otf }
4268     [ cmap = UniGB-UTF16-H, BoldFont = FandolHei-Bold.otf ]
4269     \setCJKfamilyfont { zhfs } { FandolFang-Regular.otf }
4270     [ cmap = UniGB-UTF16-H ]
4271     \setCJKfamilyfont { zhkai } { FandolKai-Regular.otf }
4272     [ cmap = UniGB-UTF16-H ]
4273     \ctex_punct_set:n { fandol }
4274     \ctex_punct_map_family:nn { \CJKrmdefault } { zhsong }
4275     \ctex_punct_map_family:nn { \CJKsfdefault } { zhhei }
4276     \ctex_punct_map_family:nn { \CJKttdefault } { zhfs }
4277     \ctex_punct_map_bfseries:nn { \CJKrmdefault, zhsong } { zhsongb }
4278     \ctex_punct_map_bfseries:nn { \CJKsfdefault, zhhei } { zhheib }
4279     \ctex_punct_map_itshape:nn { \CJKrmdefault } { zhkai }
4280   }
4281   {
4282     \ctex_load_zhmap:nnnn { rm } { zhhei } { zhfs } { zhfandolfonts }
4283     \ctex_punct_set:n { fandol }
4284     \ctex_punct_map_family:nn { \CJKrmdefault } { zhsong }
4285     \ctex_punct_map_bfseries:nn { \CJKrmdefault } { zhhei }
4286     \ctex_punct_map_itshape:nn { \CJKrmdefault } { zhkai }
4287   }
4288   { \ctex_fontset_error:n { fandol } }
4289 }
4290 {
4291   \ctex_set_upfonts:nnnnnn
4292   { FandolSong-Regular.otf }
4293   { FandolSong-Bold.otf }
4294   { FandolKai-Regular.otf }
4295   { FandolHei-Regular.otf }
4296   { FandolHei-Bold.otf }
4297   { FandolFang-Regular.otf }
4298   \ctex_set_upfamily:nnn { zhsong } { upzhserif } { upzhserifb }
4299   \ctex_set_upfamily:nnn { zhhei } { upzhsans } { upzhsansb }
4300   \ctex_set_upfamily:nnn { zhfs } { upzhmono } {}
4301   \ctex_set_upfamily:nnn { zhkai } { upzhserifit } {}
4302 }
4303 {
4304   \setCJKmainfont { FandolSong-Regular }
4305   [
4306     Extension = .otf,
4307     BoldFont  = FandolSong-Bold,
4308     ItalicFont = FandolKai-Regular
4309   ]
4310   \setCJKsansfont { FandolHei-Regular }
4311   [ Extension = .otf, BoldFont = FandolHei-Bold ]
4312   \setCJKmonofont { FandolFang-Regular }
4313   [ Extension = .otf ]
4314   \setCJKfamilyfont { zhsong } { FandolSong-Regular }

```

```

4315     [ Extension = .otf, BoldFont = FandolSong-Bold ]
4316     \setCJKfamilyfont { zhhei } { FandolHei-Regular }
4317     [ Extension = .otf, BoldFont = FandolHei-Bold ]
4318     \setCJKfamilyfont { zhfs } { FandolFang-Regular }
4319     [ Extension = .otf ]
4320     \setCJKfamilyfont { zhkai } { FandolKai-Regular }
4321     [ Extension = .otf ]
4322 }
4323 </fandol>

```

14.15.2.3 founder

```

4324 <*founder>
4325 \ctex_fontset_case:nnn
4326 {
4327     \ctex_zhmap_case:nnn
4328     {
4329         \setCJKmainfont { FZSSK.TTF }
4330         [ BoldFont = FZXBSK.TTF, ItalicFont = FZKTK.TTF ]
4331         \setCJKsansfont { FZXH1K.TTF } [ BoldFont = FZHTK.TTF ]
4332         \setCJKmonofont { FZFSK.TTF }
4333         \setCJKfamilyfont { zhsong } { FZSSK.TTF } [ BoldFont = FZXBSK.TTF ]
4334         \setCJKfamilyfont { zhhei } { FZHTK.TTF }
4335         \setCJKfamilyfont { zhkai } { FZKTK.TTF }
4336         \setCJKfamilyfont { zhfs } { FZFSK.TTF }
4337         \setCJKfamilyfont { zhli } { FZLSK.TTF }
4338         \setCJKfamilyfont { zhyou } { FZY1K.TTF } [ BoldFont = FZY3K.TTF ]
4339         \ctex_punct_set:n { founder }
4340         \ctex_punct_map_family:nn { \CJKrmdefault } { zhsong }
4341         \ctex_punct_map_family:nn { \CJKsfdefault } { zhheil }
4342         \ctex_punct_map_family:nn { \CJKttdefault } { zhfs }
4343         \ctex_punct_map_itshape:nn { \CJKrmdefault } { zhkai }
4344         \ctex_punct_map_bfseries:nn { \CJKrmdefault, zhsong } { zhsongb }
4345         \ctex_punct_map_bfseries:nn { \CJKsfdefault } { zhhei }
4346         \ctex_punct_map_bfseries:nn { zhyou } { zhyoub }
4347     }
4348     {
4349         \ctex_load_zhmap:nnnn { rm } { zhhei } { zhfs } { zhfounderfonts }
4350         \ctex_punct_set:n { founder }
4351         \ctex_punct_map_family:nn { \CJKrmdefault } { zhsong }
4352         \ctex_punct_map_bfseries:nn { \CJKrmdefault } { zhhei }
4353         \ctex_punct_map_itshape:nn { \CJKrmdefault } { zhkai }
4354     }
4355     { \ctex_fontset_error:n { founder } }
4356 }
4357 {
4358     \ctex_set_upfonts:nnnnnn
4359     { FZSSK.TTF }
4360     { FZXBSK.TTF }
4361     { FZKTK.TTF }
4362     { FZXH1K.TTF }
4363     { FZHTK.TTF }
4364     { FZFSK.TTF }
4365     \ctex_set_upfamily:nnn { zhsong } { upzhserif } { upzhserifb }
4366     \ctex_set_upfamily:nnn { zhhei } { upzhsans } { upzhsansb }
4367     \ctex_set_upfamily:nnn { zhfs } { upzhmono } {}
4368     \ctex_set_upfamily:nnn { zhkai } { upzhserifit } {}
4369     \ctex_set_upfamily:nnn { zhli } { upschrm } {}
4370     \ctex_set_upfamily:nnn { zhyou } { upschgt } {}
4371     \ctex_set_upmap:nnn { upstsl } { FZLSK.TTF } {}
4372     \ctex_set_upmap:nnn { upstht } { FZY1K.TTF } {}
4373 }
4374 {
4375     \setCJKmainfont { FZShuSong-Z01 }
4376     [ BoldFont = FZXiaoBiaoSong-B05, ItalicFont = FZKai-Z03 ]
4377     \setCJKsansfont { FZXiHeiI-Z08 } [ BoldFont = FZHei-B01 ]
4378     \setCJKmonofont { FZFangSong-Z02 }

```

```

4379 \setCJKfamilyfont { zhsong } { FZShuSong-Z01 }
4380 [ BoldFont = FZXiaoBiaoSong-B05 ]
4381 \setCJKfamilyfont { zhhei } { FZHei-B01 }
4382 \setCJKfamilyfont { zhkai } { FZKai-Z03 }
4383 \setCJKfamilyfont { zhfs } { FZFangSong-Z02 }
4384 \setCJKfamilyfont { zhli } { FZLiShu-S01 }
4385 \setCJKfamilyfont { zhyou } { FZXiYuan-M01 }
4386 [ BoldFont = FZZhunYuan-M02 ]
4387 }
4388 </founder>

```

14.15.2.4 mac 相关

按 [Issue 351](#) 的讨论，以 El Capitan 为分界，分别设置 macold (El Capitan 之前) 和 macnew (El Capitan 及之后)。检测方式则以 El Capitan 及之后的苹方字体为准。

```

4389 <*mac>
4390 \file_if_exist:nTF { /System/Library/Fonts/PingFang.ttc }
4391 { \ctex_file_input:n { ctex-fontset-macnew.def } }
4392 { \ctex_file_input:n { ctex-fontset-macold.def } }
4393 </mac>

```

macold 的设置参考了 [OS X Mavericks \(10.9\)](#) 预装的主要简体中文字体列表。

macnew 在默认字体设置方面，引入了多字重的宋体作为罗马字族，以及引入了苹方黑体作为无衬线字族。由于 Songti SC Light 的字重与 STSong 及 Windows 上的 SimSun 更接近，故默认字重使用 Songti SC Light，而不带后缀的正常字重事实上没有使用。黑体、圆体等设置也有类似的情况。

```

4394 <*macold|macnew>
4395 \ctex_fontset_case:nnnn
4396 { \ctex_fontset_error:n { mac } }
4397 <*macold>
4398 { \ctex_fontset_error:n { macold } }
4399 { \ctex_fontset_error:n { macold } }
4400 </macold>
4401 <*macnew>
4402 {
4403   \ctex_zhmap_case:nnn
4404   {
4405     \setCJKmainfont { :3:Songti.ttc }
4406     [
4407       BoldFont      = :1:Songti.ttc,
4408       ItalicFont    = :0:Kaiti.ttc,
4409       BoldItalicFont = :3:Kaiti.ttc,
4410     ]
4411     \setCJKsansfont { :2:PingFang.ttc } [ BoldFont = :8:PingFang.ttc ]
4412     \setCJKmonofont { STFANGSO.ttf }
4413     \setCJKfamilyfont { zhsong } { :3:Songti.ttc } [ BoldFont = :1:Songti.ttc ]
4414     \setCJKfamilyfont { zhhei } { :2:PingFang.ttc } [ BoldFont = :8:PingFang.ttc ]
4415     \setCJKfamilyfont { zhkai } { :0:Kaiti.ttc } [ BoldFont = :3:Kaiti.ttc ]
4416     \setCJKfamilyfont { zhfs } { STFANGSO.ttf }
4417     \setCJKfamilyfont { zhli } { :0:Baoli.ttc }
4418     \setCJKfamilyfont { zhyou } { :4:Yuanti.ttc } [ BoldFont = :0:Yuanti.ttc ]
4419     \ctex_punct_set:n { mac }
4420     \ctex_punct_map_family:nn { \CJKrmdefault } { zhsong }
4421     \ctex_punct_map_family:nn { \CJKsfdefault } { zhheil }
4422     \ctex_punct_map_family:nn { \CJKttdefault } { zhfs }
4423     \ctex_punct_map_itshape:nn { \CJKrmdefault } { zhkai }
4424     \ctex_punct_map_bfseries:nn { \CJKrmdefault, zhsong } { zhsongb }
4425     \ctex_punct_map_bfseries:nn { \CJKsfdefault } { zhhei }
4426     \ctex_punct_map_bfseries:nn { zhyou } { zhyoub }
4427   }
4428   {
4429     \ctex_load_zhmap:nnnn { rm } { zhhei } { zhfs } { zhmacfonts }
4430     \ctex_punct_set:n { mac }

```



```

4431 \ctex_punct_map_family:nn { \CJKrmdefault } { zhsong }
4432 \ctex_punct_map_bfseries:nn { \CJKrmdefault } { zhhei }
4433 \ctex_punct_map_itshape:nn { \CJKrmdefault } { zhkai }
4434 }
4435 { \ctex_fontset_error:n { macnew } }
4436 }
4437 {
4438 \ctex_set_upmap:nnn { upserif } { :3:Songti.ttc } { :1:Songti.ttc }
4439 \ctex_set_upmap:nnn { upserifit } { :0:Kaiti.ttc } { }
4440 \ctex_set_upmap:nnn { upstsl } { :0:Baoli.ttc } { }
4441 \ctex_set_upmap:nnn { upstht } { :4:Yuantti.ttc } { }
4442 \ctex_set_upmap_unicode:nnn { upsans } { :2:PingFang.ttc } { :8:PingFang.ttc }
4443 \ctex_set_upmap_unicode:nnn { upmono } { STFANGSO.ttf } { }
4444 \ctex_set_upfamily:nnn { zhsong } { upzhserif } { upzhserifb }
4445 \ctex_set_upfamily:nnn { zhhei } { upzhsans } { upzhsansb }
4446 \ctex_set_upfamily:nnn { zhfs } { upzhmono } { }
4447 \ctex_set_upfamily:nnn { zhkai } { upzhserifit } { }
4448 \ctex_set_upfamily:nnn { zhli } { upschrm } { }
4449 \ctex_set_upfamily:nnn { zhyou } { upschgt } { }
4450 }
4451 </macnew>
4452 {
4453 <*macold>
4454 \setCJKmainfont { STSong }
4455 [ BoldFont = STHeiti, ItalicFont = STKaiti ]
4456 \setCJKsansfont { STXihei } [ BoldFont = STHeiti ]
4457 \setCJKmonofont { STFangsong }
4458 \setCJKfamilyfont { zhsong } { STSong }
4459 \setCJKfamilyfont { zhhei } { STHeiti }
4460 \setCJKfamilyfont { zhfs } { STFangsong }
4461 \setCJKfamilyfont { zhkai } { STKaiti }
4462 </macold>
4463 <*macnew>
4464 \setCJKmainfont { Songti~SC~Light }
4465 [
4466 BoldFont = Songti~SC~Bold,
4467 ItalicFont = Kaiti~SC,
4468 BoldItalicFont = Kaiti~SC~Bold
4469 ]
4470 \setCJKsansfont { PingFang~SC }
4471 \setCJKmonofont { STFangsong }
4472 \setCJKfamilyfont { zhsong } { Songti~SC~Light } [ BoldFont = Songti~SC~Bold ]
4473 \setCJKfamilyfont { zhhei } { Heiti~SC~Light } [ BoldFont = Heiti~SC~Medium ]
4474 \setCJKfamilyfont { zhpfs } { PingFang~SC }
4475 \setCJKfamilyfont { zhfs } { STFangsong }
4476 \setCJKfamilyfont { zhkai } { Kaiti~SC } [ BoldFont = Kaiti~SC~Bold ]
4477 \setCJKfamilyfont { zhli } { Baoli~SC }
4478 \setCJKfamilyfont { zhyou } { Yuanti~SC~Light } [ BoldFont = Yuanti~SC~Regular ]
4479 </macnew>
4480 }
4481 </macold|macnew>

```

14.15.2.5 ubuntu

```

4482 <*ubuntu>
4483 \ctex_fontset_case:nnnn
4484 { \ctex_fontset_error:n { ubuntu } }
4485 {
4486 \ctex_zhmap_case:nnn
4487 {
4488 \setCJKmainfont { :2:NotoSerifCJK-Regular.ttc }
4489 [ BoldFont = :2:NotoSerifCJK-Bold.ttc, ItalicFont = gkai00mp.ttf ]
4490 \setCJKsansfont { :2:NotoSansCJK-Regular.ttc }
4491 [ BoldFont = :2:NotoSansCJK-Bold.ttc ]
4492 \setCJKmonofont { :2:NotoSerifCJK-Regular.ttc }
4493 [ BoldFont = :2:NotoSerifCJK-Bold.ttc ]

```

```

4494 \setCJKfamilyfont { zhsong } { :2:NotoSerifCJK-Regular.ttc }
4495 [ BoldFont = :2:NotoSerifCJK-Bold.ttc ]
4496 \setCJKfamilyfont { zhhei } { :2:NotoSansCJK-Regular.ttc }
4497 [ BoldFont = :2:NotoSansCJK-Bold.ttc ]
4498 \setCJKfamilyfont { zhkai } { gkai00mp.ttf }
4499 \ctex_punct_set:n { fandol }
4500 \ctex_punct_map_family:nn { \CJKrmdefault } { zhsong }
4501 \ctex_punct_map_family:nn { \CJKsfdefault } { zhhei }
4502 \ctex_punct_map_bfseries:nn { \CJKrmdefault, zhsong } { zhsongb }
4503 \ctex_punct_map_bfseries:nn { \CJKsfdefault, zhhei } { zhheib }
4504 \ctex_punct_map_itshape:nn { \CJKrmdefault } { zhkai }
4505 }
4506 {
4507 \ctex_load_zhmap:nnnn { rm } { zhhei } { zhsong } { zhubuntufonts }
4508 \ctex_punct_set:n { ubuntu }
4509 \ctex_punct_map_family:nn { \CJKrmdefault } { zhsong }
4510 \ctex_punct_map_bfseries:nn { \CJKrmdefault } { zhhei }
4511 \ctex_punct_map_itshape:nn { \CJKrmdefault } { zhkai }
4512 }
4513 { \ctex_fontset_error:n { ubuntu } }
4514 }
4515 {
4516 \ctex_set_upmap_unicode:nnn { upserif }
4517 { :2:NotoSerifCJK-Regular.ttc } { :2:NotoSerifCJK-Bold.ttc }
4518 \ctex_set_upmap_unicode:nnn { upsans }
4519 { :2:NotoSansCJK-Regular.ttc } { :2:NotoSansCJK-Bold.ttc }
4520 \ctex_set_upmap_unicode:nnn { upmono }
4521 { :2:NotoSerifCJK-Regular.ttc } { }
4522 \ctex_set_upmap:nnn { upserifit } { gkai00mp.ttf } { }
4523 \ctex_set_upfamily:nnn { zhsong } { upzhserif } { upzhserifb }
4524 \ctex_set_upfamily:nnn { zhhei } { upzhsans } { upzhsansb }
4525 \ctex_set_upfamily:nnn { zhkai } { upzhserifit } { }
4526 }
4527 {
4528 \setCJKmainfont { Noto~Serif~CJK~SC } [ ItalicFont = AR~PL~KaitiM~GB ]
4529 \setCJKsansfont { Noto~Sans~CJK~SC }
4530 \setCJKmonofont { Noto~Serif~CJK~SC }
4531 \setCJKfamilyfont { zhsong } { Noto~Serif~CJK~SC }
4532 \setCJKfamilyfont { zhhei } { Noto~Sans~CJK~SC }
4533 \setCJKfamilyfont { zhkai } { AR~PL~KaitiM~GB }
4534 }
4535 </ubuntu>

```

14.15.2.6 windows

\c__ctex_msyh_suffix_tl Windows 8 以后,微软雅黑由原来的 .ttf 后缀改为 .ttc 后缀,需要加以区分。

```

4536 <*windows>
4537 \file_if_exist:nTF { \c_dollar_str WINDIR/Fonts/msyh.ttc }
4538 { \tl_const:Nn \c__ctex_msyh_suffix_tl { ttc } }
4539 {
4540 \file_if_exist:nTF { msyh.ttc }
4541 { \tl_const:Nn \c__ctex_msyh_suffix_tl { ttc } }
4542 { \tl_const:Nn \c__ctex_msyh_suffix_tl { ttf } }
4543 }
4544 \ctex_fontset_case:nnn
4545 {
4546 \ctex_zhmap_case:nnn
4547 {
4548 \ctex_punct_set:n { windows }
4549 \setCJKmainfont { simsun.ttc }
4550 [ BoldFont = simhei.ttf, ItalicFont = simkai.ttf ]
4551 \setCJKsansfont { msyh.\c__ctex_msyh_suffix_tl }
4552 [ BoldFont = msyhb.\c__ctex_msyh_suffix_tl ]
4553 \setCJKmonofont { simfang.ttf }
4554 \setCJKfamilyfont { zhsong } { simsun.ttc }
4555 \setCJKfamilyfont { zhhei } { simhei.ttf }

```



```

4556 \setCJKfamilyfont { zhfs } { simfang.ttf }
4557 \setCJKfamilyfont { zhkai } { simkai.ttf }
4558 \setCJKfamilyfont { zhyahei } { msyh.\c__ctex_msyh_suffix_tl }
4559 [ BoldFont = msyhbd.\c__ctex_msyh_suffix_tl ]
4560 \setCJKfamilyfont { zhli } { simli.ttf }
4561 \setCJKfamilyfont { zhyou } { simyou.ttf }
4562 \ctex_punct_map_family:nn { \CJKrmdefault } { zhsong }
4563 \ctex_punct_map_bfseries:nn { \CJKrmdefault } { zhhei }
4564 \ctex_punct_map_itshape:nn { \CJKrmdefault } { zhkai }
4565 \ctex_punct_map_family:nn { \CJKsfdefault } { zhyahei }
4566 \ctex_punct_map_bfseries:nn { \CJKsfdefault, zhyahei } { zhyaheib }
4567 \ctex_punct_map_family:nn { \CJKttdefault } { zhfs }
4568 }
4569 {
4570 \ctex_load_zhmap:nnnn { rm } { zhhei } { zhfs } { zhwindowsfonts }
4571 \ctex_punct_set:n { windows }
4572 \ctex_punct_map_family:nn { \CJKrmdefault } { zhsong }
4573 \ctex_punct_map_bfseries:nn { \CJKrmdefault } { zhhei }
4574 \ctex_punct_map_itshape:nn { \CJKrmdefault } { zhkai }
4575 }
4576 {
4577 \tl_set:Nn \CJKrmdefault { rm }
4578 \tl_set:Nn \CJKsfdefault { sf }
4579 \tl_set:Nn \CJKttdefault { tt }
4580 }
4581 }
4582 {
4583 \ctex_set_upfonts:nnnnnn
4584 { simsun.ttc }
4585 { simhei.ttf }
4586 { simkai.ttf }
4587 { msyh.\c__ctex_msyh_suffix_tl }
4588 { msyhbd.\c__ctex_msyh_suffix_tl }
4589 { simfang.ttf }
4590 \ctex_set_upfamily:nnn { zhsong } { upzhserif } {}
4591 \ctex_set_upfamily:nnn { zhhei } { upzhserifb } {}
4592 \ctex_set_upfamily:nnn { zhfs } { upzhmono } {}
4593 \ctex_set_upfamily:nnn { zhkai } { upzhserifit } {}
4594 \ctex_set_upfamily:nnn { zhyahei } { upzhsans } { upzhsansb }
4595 \ctex_set_upfamily:nnn { zhli } { upschrn } {}
4596 \ctex_set_upfamily:nnn { zhyou } { upschgt } {}
4597 \ctex_set_upmap:nnn { upstsl } { simli.ttf } {}
4598 \ctex_set_upmap:nnn { upstht } { simyou.ttf } {}
4599 }
4600 {
4601 \setCJKmainfont { SimSun } [ BoldFont = SimHei , ItalicFont = KaiTi ]
4602 \setCJKsansfont { Microsoft~YaHei } [ BoldFont = *~Bold ]
4603 \setCJKmonofont { FangSong }
4604 \setCJKfamilyfont { zhsong } { SimSun }
4605 \setCJKfamilyfont { zhhei } { SimHei }
4606 \setCJKfamilyfont { zhfs } { FangSong }
4607 \setCJKfamilyfont { zhkai } { KaiTi }
4608 \setCJKfamilyfont { zhyahei } { Microsoft~YaHei } [ BoldFont = *~Bold ]
4609 \setCJKfamilyfont { zhli } { LiSu }
4610 \setCJKfamilyfont { zhyou } { YouYuan }
4611 }
4612 </windows>

```

14.15.3 中文字体命令

使用 \LaTeX 编译时, `macnew` 字库中由于传统黑体(黑体-简)无法使用,我们用苹方来代替。同时 `\yahei`、`\pingfang` 命令被设置为与 `\heiti` 相同。

```

\songti
\heiti
\fangsong
\kaishu
\lishu
\youyuan
\yahei
\pingfang

```

```

4613 <!*mac>
4614 \NewDocumentCommand \songti { } { \CJKfamily { zhsong } }

```

```

4615 \NewDocumentCommand \heiti { } { \CJKfamily { zhhei } }
<lubuntu> 4616 \NewDocumentCommand \fangsong { } { \CJKfamily { zhfs } }
4617 \NewDocumentCommand \kaishu { } { \CJKfamily { zhkai } }
4618 <*windows|founder|macnew>
4619 \NewDocumentCommand \lishu { } { \CJKfamily { zhli } }
4620 \NewDocumentCommand \youyuan { } { \CJKfamily { zhyou } }
4621 </windows|founder|macnew>
<windows> 4622 \NewDocumentCommand \yahei { } { \CJKfamily { zhyahei } }
4623 <*macnew>
4624 \bool_lazy_or:nnTF
4625 { \sys_if_engine_pdftex_p: }
4626 { \sys_if_engine_uptex_p: }
4627 {
4628   \cs_new_eq:NN \yahei \heiti
4629   \cs_new_eq:NN \pingfang \heiti
4630 }
4631 {
4632   \NewDocumentCommand \yahei { } { \CJKfamily { zhpj } }
4633   \NewDocumentCommand \pingfang { } { \CJKfamily { zhpj } }
4634 }
4635 </macnew>
4636 </!mac>

4637 </fontset>

```

14.15.4 zhmetrics 的字体映射

确认 \catcode, 没有重复载入检查。

```

4638 <*zhmap>
4639 \begingroup\catcode61\catcode48\catcode32=10\relax%
4640 \catcode 35=6 % #
4641 \catcode123=1 % {
4642 \catcode125=2 % }
4643 \toks0{\endlinechar=\the\endlinechar\relax}%
4644 \toks2{\endlinechar=-1}%
4645 \def\x#1 #2 {%
4646   \toks0\expandafter{\the\toks0 \catcode#1=\the\catcode#1\relax}%
4647   \toks2\expandafter{\the\toks2 \catcode#1=#2 }%
4648 \x 13 5 % carriage return
4649 \x 32 10 % space
4650 \x 35 6 % #
4651 \x 40 12 % (
4652 \x 41 12 % )
4653 \x 45 12 % -
4654 \x 46 12 % .
4655 \x 47 12 % /
4656 \x 58 12 % :
4657 \x 60 12 % <
4658 \x 61 12 % =
4659 \x 64 11 % @
4660 \x 91 12 % [
4661 \x 93 12 % ]
4662 \x 123 1 % {
4663 \x 125 2 % }
4664 \edef\x#1{\endgroup%
4665   \edef\noexpand#1{%
4666     \the\toks0 %
4667     \let\noexpand\noexpand\noexpand#1%
4668     \noexpand\noexpand\noexpand\undefined%
4669     \noexpand\noexpand\noexpand\endinput}%
4670   \the\toks2}%
4671 \expandafter\x\csname ctex@zhmap@endinput\endcsname

\ifzhmappdf 4672 \begingroup\expandafter\endgroup
4673 \expandafter\let\csname ifzhmappdf\expandafter\endcsname\csname

```

```

4674 \expandafter\ifx\csname ifctexpdf\endcsname\relax
4675 \expandafter\ifx\csname pdfoutput\endcsname\relax
4676 \iffalse\else\ifnum\pdfoutput < 1 \iffalse\else \fi\fi
4677 \else ifctexpdf\fi
4678 \endcsname

```

\ProvidesFile 提供非 L^AT_EX 格式下的 \ProvidesFile。

```

4679 \begingroup
4680 \expandafter\ifx\csname ProvidesFile\endcsname\relax
4681 \long\def\x#1\ProvidesFile#2[#3]{%
4682 #1%
4683 \immediate\write-1{File: #2 #3}%
4684 \expandafter\xdef\csname ver@#2\endcsname{#3}}
4685 \expandafter\x%
4686 \fi
4687 \endgroup

```

文件标识信息。

```

<adobe> 4688 \ProvidesFile{zhadobefonts.tex}%
<adobe> 4689 [2020/05/02 v2.5.1 Adobe font map loader for DVIPDFMx (CTEX)]
<fandol> 4690 \ProvidesFile{zhfandolfonts.tex}%
<fandol> 4691 [2020/05/02 v2.5.1 Fandol font map loader for DVIPDFMx (CTEX)]
<founder> 4692 \ProvidesFile{zhfounderfonts.tex}%
<founder> 4693 [2020/05/02 v2.5.1 Founder font map loader for pdfTeX and DVIPDFMx (CTEX)]
<mac> 4694 \ProvidesFile{zhmacfonts.tex}%
<mac> 4695 [2020/05/02 v2.5.1 Mac font map loader for pdfTeX and DVIPDFMx (CTEX)]
<ubuntu> 4696 \ProvidesFile{zhubuntufonts.tex}%
<ubuntu> 4697 [2020/05/02 v2.5.1 Ubuntu font map loader for pdfTeX and DVIPDFMx (CTEX)]
<windows> 4698 \ProvidesFile{zhwindowsfonts.tex}%
<windows> 4699 [2020/05/02 v2.5.1 Windows font map loader for pdfTeX and DVIPDFMx (CTEX)]

```

14.15.4.1 zhadobefonts.tex

```

4700 <*adobe>
4701 \ifzhmappdf
4702 %% pdfTeX does not support OTF fonts
4703 \else
4704 \special{pdf:mapline gbk@UGBK@ UniGB-UTF16-H AdobeSongStd-Light.otf}
4705 \special{pdf:mapline gbksong@UGBK@ UniGB-UTF16-H AdobeSongStd-Light.otf}
4706 \special{pdf:mapline gbkkai@UGBK@ UniGB-UTF16-H AdobeKaitiStd-Regular.otf}
4707 \special{pdf:mapline gbkhei@UGBK@ UniGB-UTF16-H AdobeHeitiStd-Regular.otf}
4708 \special{pdf:mapline gbkfs@UGBK@ UniGB-UTF16-H AdobeFangsongStd-Regular.otf}
4709 \special{pdf:mapline cyberb@Unicode@ UniGB-UTF16-H AdobeSongStd-Light.otf}
4710 \special{pdf:mapline unisong@Unicode@ UniGB-UTF16-H AdobeSongStd-Light.otf}
4711 \special{pdf:mapline unikai@Unicode@ UniGB-UTF16-H AdobeKaitiStd-Regular.otf}
4712 \special{pdf:mapline unihei@Unicode@ UniGB-UTF16-H AdobeHeitiStd-Regular.otf}
4713 \special{pdf:mapline unifs@Unicode@ UniGB-UTF16-H AdobeFangsongStd-Regular.otf}
4714 \special{pdf:mapline gbksongsl@UGBK@ UniGB-UTF16-H AdobeSongStd-Light.otf -s .167}
4715 \special{pdf:mapline gbkkaisl@UGBK@ UniGB-UTF16-H AdobeKaitiStd-Regular.otf -s .167}
4716 \special{pdf:mapline gbkheisl@UGBK@ UniGB-UTF16-H AdobeHeitiStd-Regular.otf -s .167}
4717 \special{pdf:mapline gbkfssl@UGBK@ UniGB-UTF16-H AdobeFangsongStd-Regular.otf -s .167}
4718 \special{pdf:mapline unisongsl@Unicode@ UniGB-UTF16-H AdobeSongStd-Light.otf -s .167}
4719 \special{pdf:mapline unikaisl@Unicode@ UniGB-UTF16-H AdobeKaitiStd-Regular.otf -s .167}
4720 \special{pdf:mapline uniheisl@Unicode@ UniGB-UTF16-H AdobeHeitiStd-Regular.otf -s .167}
4721 \special{pdf:mapline unifssl@Unicode@ UniGB-UTF16-H AdobeFangsongStd-Regular.otf -s .167}
4722 \fi
4723 </adobe>

```

14.15.4.2 zhfandolfonts.tex

```

4724 <*fandol>
4725 \ifzhmappdf
4726 %% pdfTeX does not support OTF fonts
4727 \else
4728 \special{pdf:mapline gbk@UGBK@ UniGB-UTF16-H FandolSong-Regular.otf}

```

```

4729 \special{pdf:mapline gbksong@UGBK@ UniGB-UTF16-H FandolSong-Regular.otf}
4730 \special{pdf:mapline gbkkai@UGBK@ UniGB-UTF16-H FandolKai-Regular.otf}
4731 \special{pdf:mapline gbkhei@UGBK@ UniGB-UTF16-H FandolHei-Regular.otf}
4732 \special{pdf:mapline gbkfs@UGBK@ UniGB-UTF16-H FandolFang-Regular.otf}
4733 \special{pdf:mapline cyberb@Unicode@ UniGB-UTF16-H FandolSong-Regular.otf}
4734 \special{pdf:mapline unisong@Unicode@ UniGB-UTF16-H FandolSong-Regular.otf}
4735 \special{pdf:mapline unikai@Unicode@ UniGB-UTF16-H FandolKai-Regular.otf}
4736 \special{pdf:mapline unihei@Unicode@ UniGB-UTF16-H FandolHei-Regular.otf}
4737 \special{pdf:mapline unifs@Unicode@ UniGB-UTF16-H FandolFang-Regular.otf}
4738 \special{pdf:mapline gbksongsl@UGBK@ UniGB-UTF16-H FandolSong-Regular.otf -s .167}
4739 \special{pdf:mapline gbkkaisl@UGBK@ UniGB-UTF16-H FandolKai-Regular.otf -s .167}
4740 \special{pdf:mapline gbkheisl@UGBK@ UniGB-UTF16-H FandolHei-Regular.otf -s .167}
4741 \special{pdf:mapline gbkfssl@UGBK@ UniGB-UTF16-H FandolFang-Regular.otf -s .167}
4742 \special{pdf:mapline unisongsl@Unicode@ UniGB-UTF16-H FandolSong-Regular.otf -s .167}
4743 \special{pdf:mapline unikaisl@Unicode@ UniGB-UTF16-H FandolKai-Regular.otf -s .167}
4744 \special{pdf:mapline uniheisl@Unicode@ UniGB-UTF16-H FandolHei-Regular.otf -s .167}
4745 \special{pdf:mapline unifssl@Unicode@ UniGB-UTF16-H FandolFang-Regular.otf -s .167}
4746 \fi
4747 </fandol>

```

14.15.4.3 zhfounderfonts.tex

```

4748 <*founder>
4749 \ifzhmappdf
4750 \pdfmapline{=gbk@UGBK@ <FZSSK.TTF}
4751 \pdfmapline{=gbksong@UGBK@ <FZSSK.TTF}
4752 \pdfmapline{=gbkkai@UGBK@ <FZKTK.TTF}
4753 \pdfmapline{=gbkhei@UGBK@ <FZHTK.TTF}
4754 \pdfmapline{=gbkfs@UGBK@ <FZFSK.TTF}
4755 \pdfmapline{=gbkli@UGBK@ <FZLSK.TTF}
4756 \pdfmapline{=gbkyou@UGBK@ <FZY1K.TTF}
4757 \pdfmapline{=cyberb@Unicode@ <FZSSK.TTF}
4758 \pdfmapline{=unisong@Unicode@ <FZSSK.TTF}
4759 \pdfmapline{=unikai@Unicode@ <FZKTK.TTF}
4760 \pdfmapline{=unihei@Unicode@ <FZHTK.TTF}
4761 \pdfmapline{=unifs@Unicode@ <FZFSK.TTF}
4762 \pdfmapline{=unili@Unicode@ <FZLSK.TTF}
4763 \pdfmapline{=uniyou@Unicode@ <FZY1K.TTF}
4764 \pdfmapline{=gbksongsl@UGBK@ <FZSSK.TTF}
4765 \pdfmapline{=gbkkaisl@UGBK@ <FZKTK.TTF}
4766 \pdfmapline{=gbkheisl@UGBK@ <FZHTK.TTF}
4767 \pdfmapline{=gbkfssl@UGBK@ <FZFSK.TTF}
4768 \pdfmapline{=gbklisl@UGBK@ <FZLSK.TTF}
4769 \pdfmapline{=gbkyousl@UGBK@ <FZY1K.TTF}
4770 \pdfmapline{=unisongsl@Unicode@ <FZSSK.TTF}
4771 \pdfmapline{=unikaisl@Unicode@ <FZKTK.TTF}
4772 \pdfmapline{=uniheisl@Unicode@ <FZHTK.TTF}
4773 \pdfmapline{=unifssl@Unicode@ <FZFSK.TTF}
4774 \pdfmapline{=unilisl@Unicode@ <FZLSK.TTF}
4775 \pdfmapline{=uniyousl@Unicode@ <FZY1K.TTF}
4776 \else
4777 \special{pdf:mapline gbk@UGBK@ unicode FZSSK.TTF}
4778 \special{pdf:mapline gbksong@UGBK@ unicode FZSSK.TTF}
4779 \special{pdf:mapline gbkkai@UGBK@ unicode FZKTK.TTF}
4780 \special{pdf:mapline gbkhei@UGBK@ unicode FZHTK.TTF}
4781 \special{pdf:mapline gbkfs@UGBK@ unicode FZFSK.TTF}
4782 \special{pdf:mapline gbkli@UGBK@ unicode FZLSK.TTF}
4783 \special{pdf:mapline gbkyou@UGBK@ unicode FZY1K.TTF}
4784 \special{pdf:mapline cyberb@Unicode@ unicode FZSSK.TTF}
4785 \special{pdf:mapline unisong@Unicode@ unicode FZSSK.TTF}
4786 \special{pdf:mapline unikai@Unicode@ unicode FZKTK.TTF}
4787 \special{pdf:mapline unihei@Unicode@ unicode FZHTK.TTF}
4788 \special{pdf:mapline unifs@Unicode@ unicode FZFSK.TTF}
4789 \special{pdf:mapline unili@Unicode@ unicode FZLSK.TTF}
4790 \special{pdf:mapline uniyou@Unicode@ unicode FZY1K.TTF}
4791 \special{pdf:mapline gbksongsl@UGBK@ unicode FZSSK.TTF -s .167}
4792 \special{pdf:mapline gbkkaisl@UGBK@ unicode FZKTK.TTF -s .167}
4793 \special{pdf:mapline gbkheisl@UGBK@ unicode FZHTK.TTF -s .167}

```

```

4794 \special{pdf:mapline gbkfssl@UGBK@ unicode FZFSK.TTF -s .167}
4795 \special{pdf:mapline gbklisl@UGBK@ unicode FZLSK.TTF -s .167}
4796 \special{pdf:mapline gbkyousl@UGBK@ unicode FZY1K.TTF -s .167}
4797 \special{pdf:mapline unisongsl@Unicode@ unicode FZSSK.TTF -s .167}
4798 \special{pdf:mapline unikaisl@Unicode@ unicode FZKTK.TTF -s .167}
4799 \special{pdf:mapline uniheisl@Unicode@ unicode FZHTK.TTF -s .167}
4800 \special{pdf:mapline unifssl@Unicode@ unicode FZFSK.TTF -s .167}
4801 \special{pdf:mapline unilisl@Unicode@ unicode FZLSK.TTF -s .167}
4802 \special{pdf:mapline uniyousl@Unicode@ unicode FZY1K.TTF -s .167}
4803 \fi
4804 </founder>

```

14.15.4.4 zhmacfonts.tex

```

4805 <*mac>
4806 \ifzhmappdf
4807 %% pdfTeX does not support OTF fonts
4808 \else
4809 \special{pdf:mapline gbk@UGBK@ UniGB-UTF16-H :3:Songti.ttc}
4810 \special{pdf:mapline gbksong@UGBK@ UniGB-UTF16-H :3:Songti.ttc}
4811 \special{pdf:mapline gbkkai@UGBK@ UniGB-UTF16-H :0:Kaiti.ttc}
4812 \special{pdf:mapline gbkhei@UGBK@ unicode :2:PingFang.ttc}
4813 \special{pdf:mapline gbkfs@UGBK@ unicode STFANGSO.ttf}
4814 \special{pdf:mapline gbkli@UGBK@ UniGB-UTF16-H :0:Baoli.ttc}
4815 \special{pdf:mapline gbkyou@UGBK@ UniGB-UTF16-H :4:Yuanti.ttc}
4816 \special{pdf:mapline cyberb@Unicode@ UniGB-UTF16-H :3:Songti.ttc}
4817 \special{pdf:mapline unisong@Unicode@ UniGB-UTF16-H :3:Songti.ttc}
4818 \special{pdf:mapline unikai@Unicode@ UniGB-UTF16-H :0:Kaiti.ttc}
4819 \special{pdf:mapline unihei@Unicode@ unicode :2:PingFang.ttc}
4820 \special{pdf:mapline unifs@Unicode@ unicode STFANGSO.ttf}
4821 \special{pdf:mapline unili@Unicode@ UniGB-UTF16-H :0:Baoli.ttc}
4822 \special{pdf:mapline uniyou@Unicode@ UniGB-UTF16-H :4:Yuanti.ttc}
4823 \special{pdf:mapline gbksongsl@UGBK@ UniGB-UTF16-H :3:Songti.ttc -s .167}
4824 \special{pdf:mapline gbkkaisl@UGBK@ UniGB-UTF16-H :0:Kaiti.ttc -s .167}
4825 \special{pdf:mapline gbkheisl@UGBK@ unicode :2:PingFang.ttc -s .167}
4826 \special{pdf:mapline gbkfssl@UGBK@ unicode STFANGSO.ttf -s .167}
4827 \special{pdf:mapline gbklisl@UGBK@ UniGB-UTF16-H :0:Baoli.ttc -s .167}
4828 \special{pdf:mapline gbkyousl@UGBK@ UniGB-UTF16-H :4:Yuanti.ttc -s .167}
4829 \special{pdf:mapline unisongsl@Unicode@ UniGB-UTF16-H :3:Songti.ttc -s .167}
4830 \special{pdf:mapline unikaisl@Unicode@ UniGB-UTF16-H :0:Kaiti.ttc -s .167}
4831 \special{pdf:mapline uniheisl@Unicode@ unicode :2:PingFang.ttc -s .167}
4832 \special{pdf:mapline unifssl@Unicode@ unicode STFANGSO.ttf -s .167}
4833 \special{pdf:mapline unilisl@Unicode@ UniGB-UTF16-H :0:Baoli.ttc -s .167}
4834 \special{pdf:mapline uniyousl@Unicode@ UniGB-UTF16-H :4:Yuanti.ttc -s .167}
4835 \fi
4836 </mac>

```

14.15.4.5 zhubuntufonts.tex

```

4837 <*ubuntu>
4838 \ifzhmappdf
4839 %% pdfTeX does not support OTF fonts
4840 \else
4841 \special{pdf:mapline gbk@UGBK@ unicode :2:NotoSerifCJK-Regular.ttc}
4842 \special{pdf:mapline gbksong@UGBK@ unicode :2:NotoSerifCJK-Regular.ttc}
4843 \special{pdf:mapline gbkkai@UGBK@ unicode gkai00mp.ttf}
4844 \special{pdf:mapline gbkhei@UGBK@ unicode :2:NotoSansCJK-Regular.ttc}
4845 \special{pdf:mapline gbkfs@UGBK@ unicode :2:NotoSerifCJK-Regular.ttc}
4846 \special{pdf:mapline cyberb@Unicode@ unicode :2:NotoSerifCJK-Regular.ttc}
4847 \special{pdf:mapline unisong@Unicode@ unicode :2:NotoSerifCJK-Regular.ttc}
4848 \special{pdf:mapline unikai@Unicode@ unicode gkai00mp.ttf}
4849 \special{pdf:mapline unihei@Unicode@ unicode :2:NotoSansCJK-Regular.ttc}
4850 \special{pdf:mapline unifs@Unicode@ unicode :2:NotoSerifCJK-Regular.ttc}
4851 \special{pdf:mapline gbksongsl@UGBK@ unicode :2:NotoSerifCJK-Regular.ttc -s .167}
4852 \special{pdf:mapline gbkkaisl@UGBK@ unicode gkai00mp.ttf -s .167}
4853 \special{pdf:mapline gbkheisl@UGBK@ unicode :2:NotoSansCJK-Regular.ttc -s .167}
4854 \special{pdf:mapline gbkfssl@UGBK@ unicode :2:NotoSerifCJK-Regular.ttc -s .167}
4855 \special{pdf:mapline unisongsl@Unicode@ unicode :2:NotoSerifCJK-Regular.ttc -s .167}

```



```

4856 \special{pdf:mapline unikaisl@Unicode@ unicode gkai00mp.ttf -s .167}
4857 \special{pdf:mapline uniheisl@Unicode@ unicode :2:NotoSansCJK-Regular.ttc -s .167}
4858 \special{pdf:mapline unifssl@Unicode@ unicode :2:NotoSerifCJK-Regular.ttc -s .167}
4859 \fi
4860 </ubuntu>

```

14.15.4.6 zhwindowsfonts.tex

```

4861 <*windows>
4862 \ifzhmappdf
4863 \pdfmapline{=gbk@UGBK@ <simsum.ttc}
4864 \pdfmapline{=gbksong@UGBK@ <simsum.ttc}
4865 \pdfmapline{=gbkkai@UGBK@ <simkai.ttf}
4866 \pdfmapline{=gbkhei@UGBK@ <simhei.ttf}
4867 \pdfmapline{=gbkfs@UGBK@ <simfang.ttf}
4868 \pdfmapline{=gbkli@UGBK@ <simli.ttf}
4869 \pdfmapline{=gbkyou@UGBK@ <simyou.ttf}
4870 \pdfmapline{=cyberb@Unicode@ <simsum.ttc}
4871 \pdfmapline{=unisong@Unicode@ <simsum.ttc}
4872 \pdfmapline{=unikai@Unicode@ <simkai.ttf}
4873 \pdfmapline{=unihei@Unicode@ <simhei.ttf}
4874 \pdfmapline{=unifs@Unicode@ <simfang.ttf}
4875 \pdfmapline{=unili@Unicode@ <simli.ttf}
4876 \pdfmapline{=uniyou@Unicode@ <simyou.ttf}
4877 \pdfmapline{=gbksongsl@UGBK@ <simsum.ttc}
4878 \pdfmapline{=gbkkaisl@UGBK@ <simkai.ttf}
4879 \pdfmapline{=gbkheisl@UGBK@ <simhei.ttf}
4880 \pdfmapline{=gbkfssl@UGBK@ <simfang.ttf}
4881 \pdfmapline{=gbklisl@UGBK@ <simli.ttf}
4882 \pdfmapline{=gbkyousl@UGBK@ <simyou.ttf}
4883 \pdfmapline{=unisongsl@Unicode@ <simsum.ttc}
4884 \pdfmapline{=unikaisl@Unicode@ <simkai.ttf}
4885 \pdfmapline{=uniheisl@Unicode@ <simhei.ttf}
4886 \pdfmapline{=unifssl@Unicode@ <simfang.ttf}
4887 \pdfmapline{=unilisl@Unicode@ <simli.ttf}
4888 \pdfmapline{=uniyousl@Unicode@ <simyou.ttf}
4889 \else
4890 \special{pdf:mapline gbk@UGBK@ unicode :0:simsum.ttc -v 50}
4891 \special{pdf:mapline gbksong@UGBK@ unicode :0:simsum.ttc -v 50}
4892 \special{pdf:mapline gbkkai@UGBK@ unicode simkai.ttf -v 70}
4893 \special{pdf:mapline gbkhei@UGBK@ unicode simhei.ttf -v 150}
4894 \special{pdf:mapline gbkfs@UGBK@ unicode simfang.ttf -v 50}
4895 \special{pdf:mapline gbkli@UGBK@ unicode simli.ttf -v 150}
4896 \special{pdf:mapline gbkyou@UGBK@ unicode simyou.ttf -v 60}
4897 \special{pdf:mapline cyberb@Unicode@ unicode :0:simsum.ttc -v 50}
4898 \special{pdf:mapline unisong@Unicode@ unicode :0:simsum.ttc -v 50}
4899 \special{pdf:mapline unikai@Unicode@ unicode simkai.ttf -v 70}
4900 \special{pdf:mapline unihei@Unicode@ unicode simhei.ttf -v 150}
4901 \special{pdf:mapline unifs@Unicode@ unicode simfang.ttf -v 50}
4902 \special{pdf:mapline unili@Unicode@ unicode simli.ttf -v 150}
4903 \special{pdf:mapline uniyou@Unicode@ unicode simyou.ttf -v 60}
4904 \special{pdf:mapline gbksongsl@UGBK@ unicode :0:simsum.ttc -v 50 -s .167}
4905 \special{pdf:mapline gbkkaisl@UGBK@ unicode simkai.ttf -v 70 -s .167}
4906 \special{pdf:mapline gbkheisl@UGBK@ unicode simhei.ttf -v 150 -s .167}
4907 \special{pdf:mapline gbkfssl@UGBK@ unicode simfang.ttf -v 50 -s .167}
4908 \special{pdf:mapline gbklisl@UGBK@ unicode simli.ttf -v 150 -s .167}
4909 \special{pdf:mapline gbkyousl@UGBK@ unicode simyou.ttf -v 60 -s .167}
4910 \special{pdf:mapline unisongsl@Unicode@ unicode :0:simsum.ttc -v 50 -s .167}
4911 \special{pdf:mapline unikaisl@Unicode@ unicode simkai.ttf -v 70 -s .167}
4912 \special{pdf:mapline uniheisl@Unicode@ unicode simhei.ttf -v 150 -s .167}
4913 \special{pdf:mapline unifssl@Unicode@ unicode simfang.ttf -v 50 -s .167}
4914 \special{pdf:mapline unilisl@Unicode@ unicode simli.ttf -v 150 -s .167}
4915 \special{pdf:mapline uniyousl@Unicode@ unicode simyou.ttf -v 60 -s .167}
4916 \fi
4917 </windows>
4918 \ctex@zhmap@endinput
4919 </zhmap>

```

14.15.5 制作 spa 文件

我们通过 XeTeX 的 \XeTeXglyphbounds 取得字体中标点符号的边界信息,为 CJKpunct 宏包制作 spa。

```

4920 <*spa>
4921 <*macro>
4922 \input expl3-generic %
4923 \ExplSyntaxOn
4924 \sys_if_engine_xetex:F
4925 {
4926   \msg_new:nnn { ctex } { xetex }
4927   { XeTeX~is~required~to~compile~this~document! }
4928   \msg_fatal:nn { ctex } { xetex }
4929 }

```

CJKpunct 定义的标点符号是:

‘ “ 「 『 (([{ < < 《 [【
 —…、。、. . : ; ! ? %))] } } } 》 』 ’ ”] 』

注意顺序不能改变。

```

4930 \seq_const_from_clist:Nn \c__ctex_punct_seq
4931 {
4932   "2018 , "201C , "300C , "300E , "3014 , "FF08 , "FF3B , "FF5B ,
4933   "3008 , "300A , "3016 , "3010 ,
4934   "2014 , "2026 , "3001 , "3002 , "FF0C , "FF0E , "FF1A , "FF1B ,
4935   "FF01 , "FF1F , "FF05 , "3015 , "FF09 , "FF3D , "FF5D , "3009 ,
4936   "300B , "3017 , "3011 , "2019 , "201D , "300D , "300F
4937 }

```

\ctex_make_spa:nn #1 是 spa 文件名,#2 是由 CJK 族名与字体构成的逗号列表。

```

4938 \cs_new_protected:Npn \ctex_make_spa:nn #1#2
4939 {
4940   \iow_open:Nn \g__ctex_spa_iow {#1}
4941   \clist_map_inline:nn {#2}
4942   { \__ctex_write_family:nn ##1 }
4943   \iow_close:N \g__ctex_spa_iow
4944 }
4945 \iow_new:N \g__ctex_spa_iow
4946 \cs_new_eq:NN \MAKESPA \ctex_make_spa:nn
4947 \cs_new_protected:Npn \__ctex_write_family:nn #1#2
4948 {
4949   \group_begin:
4950   \tex_font:D \l__ctex_punct_font = "#2" ~ at ~ 100 pt \scan_stop:
4951   \l__ctex_punct_font
4952   \clist_clear:N \l__ctex_punct_bounds_clist
4953   \seq_map_inline:Nn \c__ctex_punct_seq
4954   { \exp_args:No \__ctex_save_bounds:n { \int_use:N \tex_XeTeXcharglyph:D ##1 } }
4955   \iow_now:Nx \g__ctex_spa_iow
4956   {
4957     \token_to_str:N \ctexspadef {#1}

```

最后这三个逗号对 CJKpunct 来说是必要的。

```

4958     { \l__ctex_punct_bounds_clist , , , }
4959   }
4960   \group_end:
4961 }
4962 \cs_new_protected:Npn \__ctex_save_bounds:n #1
4963 {
4964   \clist_put_right:Nx \l__ctex_punct_bounds_clist
4965   {
4966     \__ctex_calc_bounds:nn { 1 } {#1} ,
4967     \__ctex_calc_bounds:nn { 3 } {#1}

```

```

4968     }
4969 }
4970 \clist_new:N \l__ctex_punct_bounds_clist

```

CJKpunct 要求的格式是边界空白宽度与 1em 的比值的一百倍。

```

4971 \cs_new:Npn \__ctex_calc_bounds:nn #1#2
4972 {
4973   \fp_eval:n
4974   {
4975     round
4976     (
4977       \dim_to_decimal_in_unit:nn
4978       { 100 \tex_XeTeXglyphbounds:D #1 ~ #2 }
4979       { 1 em }
4980     )
4981   }
4982 }
4983 \ExplSyntaxOff
4984 </macro>

```

下面是 CTeX 定义的一些字体。

```

4985 <*make>
4986 \input ctexpamacro %
4987 \MAKESPA {ctexpunct.spa}
4988 {
4989   {adobezhsong}      {AdobeSongStd-Light} ,
4990   {adobezhhei}       {AdobeHeitiStd-Regular} ,
4991   {adobezhkai}       {AdobeKaitiStd-Regular} ,
4992   {adobezhfs}        {AdobeFangsongStd-Regular} ,
4993   %
4994   {fandolzhsong}     {FandolSong} ,
4995   {fandolzhsongb}    {FandolSong-Bold} ,
4996   {fandolzhhei}      {FandolHei} ,
4997   {fandolzhheib}     {FandolHei-Bold} ,
4998   {fandolzhkai}      {FandolKai} ,
4999   {fandolzhfs}       {FandolFang} ,
5000   %
5001   {founderzhsong}    {FZShuSong-Z01} ,
5002   {founderzhsongb}   {FZXiaoBiaoSong-B05} ,
5003   {founderzhhei}     {FZHei-B01} ,
5004   {founderzhheil}    {FZXiHeiI-Z08} ,
5005   {founderzhkai}     {FZKai-Z03} ,
5006   {founderzhfs}      {FZFangSong-Z02} ,
5007   {founderzhli}      {FZLiShu-S01} ,
5008   {founderzhyou}     {FZXiYuan-M01} ,
5009   {founderzhyoub}    {FZZhunYuan-M02} ,
5010   %
5011   {maczhsong}        {Songti SC Light} ,
5012   {maczhsongb}       {Songti SC Bold} ,
5013   {maczhhei}         {Heiti SC Medium} ,
5014   {maczhheil}        {Heiti SC Light} ,
5015   {maczhkai}         {Kaiti SC} ,
5016   {maczhkaib}        {Kaiti SC Bold} ,
5017   {maczhfs}          {STFangsong} ,
5018   {maczhli}          {Baoli SC} ,
5019   {maczhyou}         {Yuanti SC Light} ,
5020   {maczhyoub}        {Yuanti SC Regular} ,
5021   {maczhpf}          {PingFang SC} ,
5022   {maczhpfb}         {PingFang SC Semibold} ,
5023   %
5024   {ubuntuzhsong}     {Noto Serif CJK SC} ,
5025   {ubuntuzhsongb}    {Noto Serif CJK SC Bold} ,
5026   {ubuntuzhhei}      {Noto Sans CJK SC} ,
5027   {ubuntuzhheib}     {Noto Sans CJK SC Bold} ,
5028   {ubuntuzhkai}      {AR PL KaitiM GB} ,
5029   %

```



```

5030     {windowszhsong}    {SimSun} ,
5031     {windowszhhei}     {SimHei} ,
5032     {windowszhkai}     {KaiTi} ,
5033     {windowszhfs}      {FangSong} ,
5034     {windowszhli}      {LiSu} ,
5035     {windowszhyou}     {YouYuan} ,
5036     {windowszhyawei}    {Microsoft YaHei} ,
5037     {windowszhyawei}    {Microsoft YaHei Bold}
5038   }
5039 \primitive\end
5040 </make>
5041 </spa>

```

14.16 translator 宏包的中文字典

```
5042 <*dict>
```

包括 ChineseGBK 和 ChineseUTF8 两种形式, 目前只翻译 beamer 宏包需要的定理环境名称。

```

5043 <*theorem>
5044 \providetranslation{Comments}{评论}
5045 \providetranslation{comments}{评论}
5046 \providetranslation{Comment}{评论}
5047 \providetranslation{comment}{评论}
5048 \providetranslation{Corollaries}{推论}
5049 \providetranslation{corollaries}{推论}
5050 \providetranslation{Corollary}{推论}
5051 \providetranslation{corollary}{推论}
5052 \providetranslation{Definitions}{定义}
5053 \providetranslation{definitions}{定义}
5054 \providetranslation{Definition}{定义}
5055 \providetranslation{definition}{定义}
5056 \providetranslation{Examples}{例}
5057 \providetranslation{examples}{例}
5058 \providetranslation{Example}{例}
5059 \providetranslation{example}{例}
5060 \providetranslation{Exercises}{练习}
5061 \providetranslation{exercises}{练习}
5062 \providetranslation{Exercise}{练习}
5063 \providetranslation{exercise}{练习}
5064 \providetranslation{Facts}{事实}
5065 \providetranslation{facts}{事实}
5066 \providetranslation{Fact}{事实}
5067 \providetranslation{fact}{事实}
5068 \providetranslation{Key Lemmas}{关键引理}
5069 \providetranslation{key lemmas}{关键引理}
5070 \providetranslation{Key Lemma}{关键引理}
5071 \providetranslation{key lemma}{关键引理}
5072 \providetranslation{Key Observations}{关键观察}
5073 \providetranslation{key observations}{关键观察}
5074 \providetranslation{Key Observation}{关键观察}
5075 \providetranslation{key observation}{关键观察}
5076 \providetranslation{Lemmas}{引理}
5077 \providetranslation{lemmas}{引理}
5078 \providetranslation{Lemma}{引理}
5079 \providetranslation{lemma}{引理}
5080 \providetranslation{Main Theorems}{主要定理}
5081 \providetranslation{main theorems}{主要定理}
5082 \providetranslation{Main Theorem}{主要定理}
5083 \providetranslation{main theorem}{主要定理}
5084 \providetranslation{Observations}{观察}
5085 \providetranslation{observations}{观察}
5086 \providetranslation{Observation}{观察}
5087 \providetranslation{observation}{观察}
5088 \providetranslation{Problems}{问题}

```

```

5089 \providetranslation{problems}{问题}
5090 \providetranslation{Problem}{问题}
5091 \providetranslation{problem}{问题}
5092 \providetranslation{Proofs}{证明}
5093 \providetranslation{proofs}{证明}
5094 \providetranslation{Proof}{证明}
5095 \providetranslation{proof}{证明}
5096 \providetranslation{Proof Sketch}{证明提要}
5097 \providetranslation{Proof sketch}{证明提要}
5098 \providetranslation{proof sketch}{证明提要}
5099 \providetranslation{Proof Sketches}{证明提要}
5100 \providetranslation{Proof sketches}{证明提要}
5101 \providetranslation{proof sketches}{证明提要}
5102 \providetranslation{Sketch of Proof}{证明提要}
5103 \providetranslation{Sketch of Proofs}{证明提要}
5104 \providetranslation{Sketch of proof}{证明提要}
5105 \providetranslation{Sketch of proofs}{证明提要}
5106 \providetranslation{sketch of proof}{证明提要}
5107 \providetranslation{sketch of proofs}{证明提要}
5108 \providetranslation{Propositions}{命题}
5109 \providetranslation{propositions}{命题}
5110 \providetranslation{Proposition}{命题}
5111 \providetranslation{proposition}{命题}
5112 \providetranslation{Remarks}{注}
5113 \providetranslation{remarks}{注}
5114 \providetranslation{Remark}{注}
5115 \providetranslation{remark}{注}
5116 \providetranslation{Solutions}{解}
5117 \providetranslation{solutions}{解}
5118 \providetranslation{Solution}{解}
5119 \providetranslation{solution}{解}
5120 \providetranslation{Theorems}{定理}
5121 \providetranslation{theorems}{定理}
5122 \providetranslation{Theorem}{定理}
5123 \providetranslation{theorem}{定理}
5124 </theorem>
5125 </dict>

```

14.17 ctexcap 宏包

```
5126 <*ctexcap>
```

ctexcap 是过时宏包。

```

5127 \clist_new:N \l__ctex_ctexcap_options_clist
5128 \clist_set:Nx \l__ctex_ctexcap_options_clist
5129 { \exp_not:v { opt@ \@currname . \@currentx } , heading }
5130 \msg_new:nnn { ctexcap } { deprecated }
5131 {
5132   Package~`ctexcap'~is~deprecated.\\
5133   Please~use~package~`ctex'~with~option~`#1'~instead: \\\
5134   \iow_indent:n { \token_to_str:N \usepackage [#1] \{ ctex \} } \\
5135 }
5136 \msg_warning:nnx { ctexcap } { deprecated }
5137 { \clist_use:Nn \l__ctex_ctexcap_options_clist { , ~ } }

```

ctexcap 是默认打开 heading 选项的 ctex。

```

5138 \PassOptionsToPackage { heading = true } { ctexcap }
5139 \RequirePackageWithOptions { ctex }
5140 </ctexcap>

```

14.18 ctexhook 宏包

```
5141 <*ctexhook>
```

实现 **etoolbox** 宏包的 `\AtEndPreamble` 和 `\AfterEndPreamble`。

```

5142 \cs_new_protected:Npn \ctex_at_end_preamble:n
5143   { \tl_gput_right:Nn \g__ctex_end_preamble_hook_tl }
5144 \cs_new_protected:Npn \ctex_after_end_preamble:n
5145   { \tl_gput_right:Nn \g__ctex_after_end_preamble_hook_tl }
5146 \cs_new_protected:Npn \CTEX@document@left@hook
5147   { \group_end: \g__ctex_end_preamble_hook_tl \group_begin: }
5148 \cs_new_protected:Npn \CTEX@document@right@hook
5149   { \scan_stop: \g__ctex_after_end_preamble_hook_tl \tex_ignorespaces:D }
5150 \cs_set_nopar:Npx \document
5151   {
5152     \CTEX@document@left@hook
5153     \exp_not:o { \document }
5154     \CTEX@document@right@hook
5155   }
5156 \tl_new:N \g__ctex_end_preamble_hook_tl
5157 \tl_new:N \g__ctex_after_end_preamble_hook_tl

```

如果宏包已经被载入,则钩子无效,给出警告。在 **filehook** 的钩子之前起作用。

```

5158 \cs_new_protected:Npn \ctex_at_begin_package:nn #1
5159   {
5160     \@ifpackageloaded {#1}
5161       { \__ctex_package_loaded_warning:nn {#1} }
5162       { \ctex_gadd_package_hook:nnn { begin } {#1} }
5163   }
5164 \cs_new_protected:Npn \__ctex_package_loaded_warning:nn #1#2
5165   { \msg_warning:nnx { ctex } { invalid-hook } {#1} }
5166 \msg_new:nnn { ctex } { invalid-hook }
5167   {
5168     Package~`#1'~is~loaded. ~~
5169     \token_to_str:N \ctex_at_begin_package:nn \{#1\}\{...\}~is~invalid.
5170   }

```

与 **filehook** 的 `\AtEndOfPackageFile*` 类似,如果原来没有在载入宏包则在宏包末尾执行语句,否则立即执行。在 **filehook** 的钩子之后起作用。

```

5171 \cs_new_protected:Npn \ctex_at_end_package:nn #1
5172   {
5173     \@ifpackageloaded {#1}
5174       { \use:n }
5175       { \ctex_gadd_package_hook:nnn { end } {#1} }
5176   }

```

给钩子附加内容。

```

5177 \cs_new_protected:Npn \ctex_gadd_package_hook:nnn #1#2
5178   { \ctex_gadd_hook:cn { g__ctex_at_ #1 _ #2 _hook_tl } }
5179 \cs_new_protected:Npn \ctex_gadd_hook:Nn #1
5180   {
5181     \tl_if_exist:NF #1 { \tl_new:N #1 }
5182     \tl_gput_right:Nn #1
5183   }
5184 \cs_generate_variant:Nn \ctex_gadd_hook:Nn { c }

```

宏包钩子,只执行一次,用后清除。

```

5185 \cs_new_protected:Npn \ctex_use_package_hook:nn #1#2
5186   {
5187     \group_begin: \exp_args:Nnc \group_end:
5188     \__ctex_use_package_hook_aux:N { g__ctex_at_ #1 _ #2 _hook_tl }
5189   }
5190 \cs_new_protected:Npn \__ctex_use_package_hook_aux:N #1
5191   { \cs_if_exist_use:NT #1 { \cs_undefine:N #1 } }

```

`\@reset@ptions` `\CTEX@reset@ptions@hook` `\@pushfilename` 内部的 `\@currname` 和 `\@currentx` 保存的是前一个宏包的状态, 不能使用。需要对其后的 `\@reset@ptions` 做补丁来实现 `\ctex_at_begin_package:nn` 的功能。

```

5192 \tl_put_right:Nn \@reset@ptions { \CTEX@reset@ptions@hook }
5193 \cs_new_protected:Npn \CTEX@reset@ptions@hook
5194 {
5195     \cs_if_eq:NNT \@currentx \@pkgextension
5196     { \ctex_use_package_hook:nn { begin } { \@currname } }
5197 }

```

`\@popfilename` `\CTEX@popfilename@hook` 对 `\@popfilename` 做补丁来实现 `\ctex_at_end_package:nn` 的功能。

```

5198 \tl_put_left:Nn \@popfilename { \CTEX@popfilename@hook }
5199 \cs_new_protected:Npn \CTEX@popfilename@hook
5200 {
5201     \cs_if_eq:NNT \@currentx \@pkgextension
5202     { \ctex_use_package_hook:nn { end } { \@currname } }
5203 }
5204 </ctexhook>

```

14.19 ctexpatch 宏包

5205 <*ctexpatch>

`\ctex_patch_cmd_once:NnnnTF` 只进行第一次匹配进行替换。参数 #2 是宏重建时的 `\catcode` 设置。

```

5206 \cs_new_protected:Npn \ctex_patch_cmd_once:NnnnTF #1#2
5207 {
5208     \ctex_patch_boot:NnnnTF \__ctex_patch_cmd:Nnnnnw #1
5209     { once } {#2} { \use_i:nn } { \use_ii:nn }
5210 }

```

`\ctex_patch_cmd_all:NnnnTF` 替换所有匹配到的文本。

```

5211 \cs_new_protected:Npn \ctex_patch_cmd_all:NnnnTF #1#2
5212 {
5213     \ctex_patch_boot:NnnnTF \__ctex_patch_cmd:Nnnnnw #1
5214     { all } {#2} { \use_i:nn } { \use_ii:nn }
5215 }

```

`\ctex_patch_cmd:Nnn` 快捷方式, 在补丁的时候关闭 \LaTeX 语法和设置 @ 为字母类, 补丁失败时给出警告。

```

5216 \cs_new_protected:Npn \ctex_patch_cmd:Nnn #1
5217 {
5218     \ctex_patch_boot:NnnTF \__ctex_patch_cmd:Nnnnnw #1
5219     { once }
5220     {
5221         \ExplSyntaxOff
5222         \char_set_catcode_letter:n { 64 }
5223     }
5224     { }
5225     { \ctex_patch_failure:N #1 }
5226 }
5227 \cs_new_protected:Npn \ctex_patch_failure:N #1
5228 { \msg_warning:nnx { ctex } { patch-failure } { \token_to_str:N #1 } }
5229 \msg_new:nnn { ctex } { patch-failure }
5230 { Oops!~Command~`#1'~is~NOT~patchable.\\ }

```

`\ctex_preto_cmd:NnnTF` 在宏的原本定义前面增加钩子。

```

5231 \cs_new_protected:Npn \ctex_preto_cmd:NnnTF #1#2
5232 {
5233     \ctex_patch_boot:NnnnTF \__ctex_hookto_cmd:Nnnnw #1
5234     { left } {#2} { \use_i:nn } { \use_ii:nn }
5235 }

```

`\ctex_appto_cmd:NnnTF` 在宏的原本定义后面追加钩子。

```
5236 \cs_new_protected:Npn \ctex_appto_cmd:NnnTF #1#2
5237 {
5238   \ctex_patch_boot:NnnTF \__ctex_hookto_cmd:Nnnw #1
5239   { right } {#2} { \use_i:nn } { \use_ii:nn }
5240 }
```

`\ctex_patch_boot:NnnTF` 参数记号 # 作为宏的参数被读入时,总是会双写,会影响随后的字符串替换。需要先将它转换为普通符号。

```
5241 \cs_new_protected:Npn \ctex_patch_boot:NnnTF #1#2#3#4#5#6
5242 {
5243   \cs_set_protected:Npx \__ctex_patch_true:w { \exp_not:n {#5} }
5244   \cs_set_protected:Npx \__ctex_patch_false:w { \exp_not:n {#6} }
5245   \group_begin:
5246     \char_set_catcode_other:n { 35 }
5247     \ctex_parse_name:NN #1 #2 {#3} {#4}
5248   }
5249   \cs_new_eq:NN \__ctex_patch_true:w \use_i:nn
5250   \cs_new_eq:NN \__ctex_patch_false:w \use_ii:nn
```

`\ctex_parse_name:NN` 用 `\DeclareRobustCommand` 定义的宏或者由 `\newcommand` 或 `\newrobustcmd` 定义的带一个可选参数的宏第一次展开的结果都不是其实际定义,实际定义被保存在另外的宏中。由这些命令定义的宏的第一次展开结果可以有下面的形式(细节可查阅 `xpatch` 的文档):

```
1 \protect_\xaa_\_          % \DeclareRobustCommand\xaa[1]{...}
2 \protect_\xab_\_          % \DeclareRobustCommand\xab[1] []{...}
3 \@protected@testopt_\xac_\_ \xac_\_ { } % \newcommand\xac[1] []{...}
4 \@testopt_\xad_\_          % \newrobustcmd\xad[1] []{...}
5 \x@protect_\1\protect_\1_\_ % \DeclareRobustCommand\1[1]{...}
6 \x@protect_\2\protect_\2_\_ % \DeclareRobustCommand\2[1] []{...}
7 \@protected@testopt_\3\3_\_ \3_\_ { } % \newcommand\3[1] []{...}
8 \@testopt_\4_\_          % \newrobustcmd\4[1] []{...}
```

`ctexpatch` 的主要原理是先对宏的 `\meaning` 作字符串替换,然后再用 `\scantokens` 来重建它。我们希望对宏的实际定义打补丁,为此需要先得到对应的名字。`letltxmacro`、`show2e` 和 `xpatch` 宏包中都有类似的工作。

```
5251 \cs_new_protected:Npn \ctex_parse_name:NN #1#2
5252 { \ctex_parse_name:Nnx #1#2 { \cs_to_str:N #2 } }
5253 \group_begin:
5254 \cs_set_protected:Npn \__ctex_tmp:w #1#2#3
5255 {
5256   \cs_new_protected:Npn \ctex_parse_name:NNn ##1##2##3
5257   {
5258     \bool_lazy_or:nnTF
5259     { \cs_if_exist_p:c { ##3 ~ } }
5260     { \cs_if_exist_p:c { #1##3 } }
5261     {
5262       \group_begin:
5263       \use:x
5264       {
5265         \group_end:
5266         \__ctex_parse_name:nNNNnN
5267         { \cs_replacement_spec:N ##2 }
5268         \exp_not:N ##2
5269         \exp_not:c { ##3 ~ }
5270         \exp_not:c { #1##3 }
5271       } { ##3 } ##1
5272     }
5273     { ##1##2 }
5274   }
5275   \cs_new_protected:Npn \__ctex_parse_name:nNNNnN ##1##2##3##4##5##6
5276   {
5277     \exp_args:Nc ##6
```

```

5278     {
5279         \str_case:nnTF {##1}
5280         {
5281             { \protect ##3 } { }
5282             { \x@protect ##2 \protect ##3 } { }
5283         }
5284         {
5285             \str_if_eq:eeTF
5286             { \exp_not:n { #1@protected@ ##3 #1##3 } }
5287             {
5288                 \exp_last_unbraced:Ne \__ctex_parse_name:w
5289                 { \cs_replacement_spec:N ##3 } #3 ~ #2 \q_stop
5290             }
5291             { #1##5 ~ } { ##5 ~ }
5292         }
5293     }
5294     \str_case:onTF { \__ctex_parse_name:w ##1 #3 ~ #2 \q_stop }
5295     {
5296         { #1@protected@ ##2 ##4 } { }
5297         { #1@ ##4 } { }
5298     }
5299     { #1##5 } {##5}
5300 }
5301 }
5302 }
5303 \cs_new:Npn \__ctex_parse_name:w ##1 #3 ~ ##2 #2 ##3 \q_stop { ##1##2 }
5304 }
5305 \use:x
5306 {
5307     \__ctex_tmp:w
5308     { \c_backslash_str }
5309     { \c_left_brace_str }
5310     { \tl_to_str:n { testopt } }
5311 }
5312 \group_end:
5313 \cs_generate_variant:Nn \ctex_parse_name:NNn { NNx }

```

```

\l__ctex_prefix_str
\l__ctex_parameter_str
\l__ctex_replacement_str

```

分别保存宏的 `\meaning` 中的前缀、参数文本和替换文本。

```

5314 \str_new:N \l__ctex_prefix_str
5315 \str_new:N \l__ctex_parameter_str
5316 \str_new:N \l__ctex_replacement_str

```

```

\ctex_get_macro_meaning:NTF
\__ctex_get_macro_meaning:w

```

解构待补丁宏的 `\meaning`。若命令不是宏，则走向 `false` 分支。

```

5317 \group_begin:
5318 \cs_set_protected:Npn \__ctex_tmp:w #1
5319 {
5320     \prg_new_protected_conditional:Npnn
5321     \ctex_get_macro_meaning:N ##1 { TF }
5322     {
5323         \exp_after:wN \__ctex_get_macro_meaning:w
5324         \token_to_meaning:N ##1 \q_mark #1 -> \q_mark \q_stop
5325     }
5326     \cs_new_protected:Npn \__ctex_get_macro_meaning:w
5327     ##1 #1 ##2 -> ##3 \q_mark ##4 \q_stop
5328     {
5329         \tl_if_empty:nTF { ##4 }
5330         { \prg_return_false: }
5331         {
5332             \str_set:Nn \l__ctex_prefix_str { ##1 }
5333             \str_set:Nn \l__ctex_parameter_str { ##2 }
5334             \str_set:Nn \l__ctex_replacement_str { ##3 }
5335             \prg_return_true:
5336         }
5337     }
5338 }

```

```

5339 \exp_args:No \__ctex_tmp:w { \tl_to_str:n { macro: } }
5340 \group_end:

```

\ctex_if_rescanable:NnTF 检查宏是否可以重建。

```

5341 \cs_new_protected:Npn \ctex_if_rescanable:NnTF #1#2#3#4
5342 {
5343   \ctex_get_macro_meaning:NnTF #1
5344   {
5345     \__ctex_patch_rebuild:Nn \__ctex_rebuild_cmd:w {#2}
5346     \cs_if_eq:NNTF #1 \__ctex_rebuild_cmd:w {#3} {#4}
5347   }
5348   {#4}
5349 }
5350 \cs_new_eq:NN \__ctex_rebuild_cmd:w \prg_do_nothing:

```

__ctex_patch_rebuild:Nn 使用 \tl_rescan:nn 来重新记号化 \meaning 字符串。

```

5351 \cs_new_protected:Npn \__ctex_patch_rebuild:Nn #1#2
5352 {
5353   \__ctex_patch_rescan:NNn \l__ctex_prefix_tl \l__ctex_prefix_str {#2}
5354   \__ctex_patch_rescan:NNn \l__ctex_parameter_tl \l__ctex_parameter_str {#2}
5355   \__ctex_patch_rescan:NNn \l__ctex_replacement_tl \l__ctex_replacement_str {#2}
5356   \use:x
5357   {
5358     \exp_not:o { \l__ctex_prefix_tl } \tex_def:D \exp_not:N #1
5359     \exp_not:o { \l__ctex_parameter_tl }
5360     { \exp_not:o { \l__ctex_replacement_tl } }
5361   }
5362 }
5363 \cs_new_protected:Npn \__ctex_patch_rescan:NNn #1#2#3
5364 {
5365   \str_if_empty:NnTF #2
5366   { \tl_clear:N #1 }
5367   { \tl_set_rescan:Nno #1 {#3} {#2} }
5368 }
5369 \tl_new:N \l__ctex_prefix_tl
5370 \tl_new:N \l__ctex_parameter_tl
5371 \tl_new:N \l__ctex_replacement_tl

```

__ctex_patch_cmd:Nnnnw 对宏的替换文本进行字符串替换,然后重建。

```

5372 \cs_new_protected:Npn \__ctex_patch_cmd:Nnnnw #1#2#3#4#5
5373 {
5374   \group_end:
5375   \ctex_if_rescanable:NnTF #1 {#3}
5376   {
5377     \use:x
5378     {
5379       \__ctex_patch_replace:nnnTF {#2}
5380       { \tl_to_str:n {#4} }
5381       { \tl_to_str:n {#5} }
5382     }
5383     {
5384       \__ctex_patch_rebuild:Nn #1 {#3}
5385       \__ctex_patch_true:w
5386     }
5387     { \__ctex_patch_false:w }
5388   }
5389   { \__ctex_patch_false:w }
5390 }

```

__ctex_patch_replace:nnnTF 替换前先检查原文本是否存在。

```

5391 \cs_new_protected:Npn \__ctex_patch_replace:nnnTF #1#2#3#4
5392 {
5393   \tl_if_in:NnTF \l__ctex_replacement_str {#2}

```

```

5394     { \use:c { tl_replace_ #1 :Nnn } \l__ctex_replacement_str {#2} {#3} #4 }
5395 }

```

`__ctex_hookto_cmd:Nnnnw` 在宏的前/后附加钩子。

```

5396 \cs_new_protected:Npn \__ctex_hookto_cmd:Nnnnw #1#2#3#4
5397 {
5398     \group_end:
5399     \ctex_get_macro_meaning:NTF #1
5400     {
5401         \str_if_empty:NTF \l__ctex_parameter_str
5402         { \__ctex_hookto_cmd_parameterless:Nnnnw }
5403         { \__ctex_hookto_cmd_parameter:Nnnnw }
5404         #1 {#2} {#3} {#4}
5405     }
5406     { \__ctex_patch_false:w }
5407 }

```

`__ctex_hookto_cmd_parameterless:Nnnnw` 如果宏没有参数,可以直接进行附加操作。注意保持宏的前缀。

```

5408 \cs_new_protected:Npn \__ctex_hookto_cmd_parameterless:Nnnnw #1#2#3#4
5409 {
5410     \str_if_empty:NF \l__ctex_prefix_str
5411     { \tl_rescan:no {#3} { \l__ctex_prefix_str } }
5412     \tex_edef:D #1
5413     {
5414         \use:c { __ctex_ #2 _hook_aux:nn }
5415         { \exp_not:o {#1} }
5416         { \exp_not:n {#4} }
5417     }
5418     \__ctex_patch_true:w
5419 }
5420 \cs_generate_variant:Nn \tl_rescan:nn { no }
5421 \cs_new:Npn \__ctex_left_hook_aux:nn #1#2 { #2#1 }
5422 \cs_new_eq:NN \__ctex_right_hook_aux:nn \use:nn

```

`__ctex_hookto_cmd_parameter:Nnnnw` 如果宏有参数,需要在字符串中进行附加,然后再重建。

```

5423 \cs_new_protected:Npn \__ctex_hookto_cmd_parameter:Nnnnw #1#2#3#4
5424 {
5425     \__ctex_patch_rebuild:Nn \__ctex_rebuild_cmd:w {#3}
5426     \cs_if_eq:NNTF #1 \__ctex_rebuild_cmd:w
5427     {
5428         \use:c { str_put_ #2 :Nn } \l__ctex_replacement_str {#4}
5429         \__ctex_patch_rebuild:Nn #1 {#3}
5430         \__ctex_patch_true:w
5431     }
5432     { \__ctex_patch_false:w }
5433 }
5434 </ctexpatch>

```


版本历史

v2.0 (2014/03/06 – 2015/05/06)

General: c5size, cs4size 是过时选项。	41
captiondelimiter 是过时选项。	116
fancyhdr 成为过时选项, 原选项功能总是打开。	43
fntef 成为过时选项, 原选项功能总是打开。	43
hyperref 成为过时选项, 原选项功能总是打开。	43
indent, noindent 是过时选项。	41
nofonts, adobefonts, winfonts 是过时选项。	42
nopunct 是过时选项。	42
nospace 是过时选项。	42
nozhmap 是过时选项。	42
punct 选项可以设置标点格式。	42
ctex 宏包新增 heading 选项。	42
\CTEXindent, \CTEXnoindent 是过时命令。	84
\CTEXsetup, \CTEXoptions 是过时命令。	15, 81
\CTEXunderdot, \CTEXunderline, \CTEXunderdblline, \CTEXunderwave, \CTEXsout, \CTEXxout 是过时命令; \CTEXfilltwosides 是过时环境。	76
标题设置新增 pagestyle 选项。	88
调整 \footnotesep 的大小, 以适合行距的变化。	115
兼容 extsizes 宏包、beamer、memoir 等提供的更多字号选项。	44
将标题汉化功能加入 ctex.sty。	85
将中文字号功能提取到可以独立使用的 ctexsize。	112
解决 etoolbox 与 breqn 关于 \end 的冲突。	116
默认关闭 CJKfntef 或 xeCJKfntef 的彩色设置。	76
删除 c19gsn.fd 和 c19gkai.fd。	1
通过 LuaTeX-jan 宏包支持 Lua [®] TeX。	51
新增 autoindent 选项。	41
新增 fontset 选项。	42
新增 linespread 选项。	41
新增 linestretch 选项。	82
新增 scheme 选项, 并将 cap 和 nocap 列为过时选项。	43
新增 zhmCJK 支持选项。	42
新增 zihao 选项。	41
新增统一设置接口 \ctexset。	81
应用 [®] TeX 重新整理代码。	1
中文字号不再采用近似值。	113
自动检测操作系统, 载入对应的字体配置。	75
\CJK@surr: 解决与 \nouppercase 的冲突。	48

v2.0.1 (2015/05/15)

General: 修复 10pt、11pt 等选项无效的问题。	44
---	----

v2.0.2 (2015/05/16)

General: 修复加载 ctex 宏包后章节标题后第一段无段首缩进的问题。	112
---	-----

v2.1 (2015/05/18 – 2015/06/19)

General: format+, nameformat+ 等带加号的选项, 加号与前面的文字之间可以有可选的空格。	87
nameformat 可以接受章节名字为参数。	86
不依赖 ifpdf 宏包。	39
不再设置 hyperref 宏包的 colorlinks 选项。	75
给 pdf [®] TeX 下的非 UTF-8 编码 CJK 字体族加上 CMap。	46

将章节标题设置功能提取到可以独立使用的宏包

ctexheading 中。	1
新的标题格式选项 aftertitle。	87
修复 ctexbook 和 ctexrep 类的中文 part/number 选项初值为空的错误。	106
\ctex_if_platform_macos:TF: 改用 /Library/Fonts/Songti.ttc 为特征文件。	117
hyperref: 补充定义 \hypersetup。	43

v2.2 (2015/06/21 – 2015/06/30)

General: before skip 和 after skip 选项的符号不再有特殊意义。	87
before skip、after skip 和 indent 选项支持表达式。	89
不再依赖 etoolbox 宏包。	1
非 ctexart 类的 part/before skip 和 part/after skip 选项有意义。	90
给 enumitem 宏包注册 \chinese 和 \zhnum。	84
将文档开头和宏包末尾钩子提取到 ctexhook 宏包中。	136
将中文版式下的 part 和 chapter 标题的 nameformat 和 titleformat 选项的初值合并到 format 中。	106
删去 etoolbox 与 breqn 的兼容补丁。	116
新的标题格式选项 afterindent。	87
新的标题格式选项 numbering。	87
新的标题格式选项 runin。	87
新增子宏包 ctexpatch 实现给宏打补丁功能。	138

v2.3 (2015/09/17 – 2016/01/05)

General: .value_required: 和 .value_forbidden: 已过时。	40
代码实现避免使用 \lowercase 技巧 (Joseph Wright)。	39
更新 LuaTeX-jan 支持 (20150922.0)。	51
更新 unicode-math 宏包补丁。	52
兼容 titleps 宏包。	101
修复 nameformat 作用域问题。	86
与 [®] TeX3 (2015/12/20) 同步。	37

v2.4 (2015/02/19 – 2016/04/25)

General: 初步支持 up [®] TeX。	72
加强 beamer 宏包支持。	1
提供 translator 宏包的中文定理名称翻译。	135
提供 up [®] TeX 的 NFSS 字体定义。	119
正确更新 CJK 包的 \CJKfamilydefault。	74
正确设置 up [®] TeX 下字体命令。	73
\ctex_parse_name:NN: 修复宏名解析错误。	139

v2.4.1 (2016/04/26 – 2016/05/14)

General: beamer 不调整默认行距。	112
beamer 不调整默认字体大小。	112
使用 bootfont.bin 判断 Windows XP 以避免权限问题。	126
随字体更新 up [®] TeX 的 \xkanjiskip。	79
新的标题格式选项 part/fixbefore skip 和 chapter/fixbefore skip。	88
正确更新 up [®] TeX 的 \CJKfamilydefault。	74
\ccwd: 正确设置 up [®] TeX 下的 \ccwd。	77
\chinese: 支持 \pagenumbering。	84
zihao: 不允许无参 zihao 选项。	41

v2.4.2	(2016/05/15)	v2.4.13	(2018/03/23)
General: 恢复 <code>luatexja</code> 对 <code>\emshape</code> 和 <code>\eminnershape</code> 的重定义。	53	<code>\CTEX@selectfont@hook</code> : 修正导言区 <code>\selectfont</code> 钩子位置。	78
<code>\em</code> : 兼容 <code>upL^AT_EX</code> 2016/05/07u00 的定义。	72	v2.4.14	(2018/05/01)
v2.4.3	(2016/06/03 – 2016/08/26)	General: 配置 <code>macnew</code> 的默认字体设置。	124
General: 更新 <code>unicode-math</code> 补丁。	59	区分 <code>macold</code> 及 <code>macnew</code> 。	124
简化 <code>fontspec</code> 补丁。	53	为 <code>macnew</code> 配置字体命令。	127
确保 <code>\proofname</code> 非空。	85	v2.4.15	(2019/01/29 – 2019/04/05)
删除选项 <code>part/fixbeforeskip</code> 和 <code>chapter/fixbeforeskip</code> 。	88	General: 定义 <code>part/hang</code> 和 <code>chapter/hang</code> 。	89
新的标题格式选项 <code>fixskip</code> 。	87	将 JY2 和 JT2 编码的字体定义提取到单独的文件中。	118
v2.4.4	(2016/09/09 – 2016/09/19)	将 <code>upL^AT_EX</code> 的默认字体由 <code>mc</code> 改为 <code>zhrm</code> , 并启用 <code>\jfam</code> 。	72
General: 不再默认设置 <code>xeCJK</code> 的伪粗体。	51	局部指定 <code>autoindent</code> 为 <code>false</code> , 并交换 <code>\CTEX@XXX@indent</code> 与 <code>\CTEX@XXX@format</code> 的顺序。	104
改进 <code>hyperref</code> 宏包的标题锚点设置。	98	同步 <code>L^AT_EX 3</code> 2019/03/05。	1
解决 <code>zhmap</code> 文件的 <code>\catcode</code> 问题。	47	统一“方正细黑一_GBK”的名称为 <code>FZXiHeiI-Z08</code> 。	123
使用 <code>titlesec</code> 时, 章节目录也使用 <code>CT_EX</code> 的编号。	101	显式补丁 <code>upL^AT_EX</code> 的 <code>\rmfamily</code> 等字体命令。	72
提供 <code>\CTEXifname</code> 。	89	修正 <code>part/indent</code> 和 <code>chapter/indent</code> 的实现方法。	89
提供 <code>\partmark</code> 。	89	v2.4.16	(2019/05/11 – 2019/05/29)
新的标题格式选项 <code>break</code> 。	87	General: 更好地兼容 <code>nameref</code> 宏包。	99
v2.4.5	(2016/10/01 – 2016/10/25)	修正 <code>part/indent</code> 和 <code>chapter/indent</code> 的实现方法, 在其标题内部禁用 <code>autoindent</code> 。	89
General: 新的标题格式选项 <code>chapter/lofskip</code> 和 <code>chapter/lotskip</code> 。	88	允许设置 <code>autoindent</code> 为 0。	39
新的标题格式选项 <code>hang</code> 。	87	v2.5	(2019/10/25 – 2020/04/30)
新的标题格式选项 <code>tocline</code> 。	87	General: <code>ubuntu</code> 改用思源(Noto CJK)和文鼎字库, 不再支持使用 <code>pdfL^AT_EX</code> 编译。	125
<code>\ps@headings</code> : 修复补丁失败。	103	标题选项 <code>format</code> 也可以接受参数。	89
v2.4.6	(2016/10/31 – 2016/11/20)	不再支持 Windows XP 系统, <code>windowsold</code> 和 <code>windowsnew</code> 成为过时字库选项。	126
General: <code>\CTEXifname</code> 初始为假。	89	不再自动载入 <code>CJKfntef</code> 或 <code>xeCJKfntef</code> 宏包。	76
支持字体属性可选项在后的新语法。	64	操作系统检测移动至载入中文字库处, 且不再需要依赖特定引擎。	75
重新初始化 <code>\ifthechapter</code> 等。	102	处理 <code>\ctex_file_input:n</code> 在 <code>ctexsize</code> 中未定义的错误。	37
v2.4.7	(2016/12/23 – 2016/12/27)	给 <code>L^AT_EX</code> 和 <code>upL^AT_EX</code> 下的文档类指定驱动为 <code>DVIPDFMx</code> 。	45
General: 依赖 <code>pxeverysel</code> 宏包。	74	更新 <code>LuaTeX-ja</code> 支持(20200412.0)。	51
<code>\ps@headings</code> : 修复 <code>ctexrep</code> 类的 <code>\chaptermark</code> 汉化错误。	103	兼容 <code>titletoc</code> 宏包。	102
v2.4.8	(2017/02/23)	删除 <code>fontspec</code> 补丁。	53
General: 解决与 <code>fontspec</code> 2017/01/24 v2.5d 的字体族匹配兼容问题。	65	使用环境变量代替绝对路径查找字体。	126
v2.4.9	(2017/02/27)	所有引擎下默认编码均设为 UTF-8。	41
General: 调整 <code>unicode-math</code> 补丁的代码顺序。	59	为 <code>macnew</code> 增加粗楷体、隶书和圆体的定义。	124
v2.4.10	(2017/07/19 – 2017/07/23)	移除 <code>\CTEXunderdot</code> 、 <code>\CTEXunderline</code> 、 <code>\CTEXunderdblline</code> 、 <code>\CTEXunderwave</code> 、 <code>\CTEXsout</code> 、 <code>\CTEXxout</code> 、 <code>CTEXfilltwosides</code> 等命令和环境。	76
General: 常数 <code>\c_minus_one</code> 已过时。	37	应用新内核中的 <code>\labelformat</code> 。	106
定义 <code>\cht</code> 、 <code>\cdp</code> 和 <code>\cwd</code> 。	79	允许 <code>macnew</code> 在 <code>L^AT_EX</code> 和 <code>upL^AT_EX</code> 下使用。	124
使用 <code>lazy</code> 函数对 Boolean 表达式进行最小化运算 (<code>L^AT_EX 3</code> 2017/07/19)。	37	在 <code>ctexsize</code> 也载入 <code>fix-cm</code> 。	37
v2.4.11	(2017/08/17 – 2017/11/21)	增加宏包开头钩子。	136
General: 不把 <code>Enclosed Alphanumerics</code> 设置为 <code>JAchar</code> 。	52	增加字体映射文件 <code>zhmacfonts.tex</code> 。	131
不把希腊和西里尔字母设置为 <code>JAchar</code> 。	52	重构标题选项 <code>indent</code> 和 <code>hang</code> 。	88
因上游 <code>l3keys</code> 变化, 重新定义 <code>format_U+</code> 等带空格加号的选项。	87	重构字体选项 <code>AlternateFont</code> 。	66
<code>\ps@headings</code> : 补充页眉空格。	103	<code>\CTEX@selectfont@hook</code> : 兼容 KOMA-Script 的 <code>\selectfont</code> 补丁。	78
v2.4.12	(2017/12/05 – 2018/01/27)	<code>\ctex_if_platform_macos:TF</code> : 改用 <code>/System/Library/Fonts/Menlo.ttc</code> 为特征文件。	117
General: 同步 <code>L^AT_EX 3</code> 2017/12/16。	1		
修正 <code>\ctexset</code> 在 <code>ctexheading</code> 包中无定义的错误(曾祥东)。	81		
正确使用 <code>\ltjsetkanjiskip</code> 和 <code>\ltjsetxkanjiskip</code> 。	72		
<code>\CTeX</code> : 不依赖 <code>\ifincsname</code> 。	116		

fntef: 仅在该选项启用时会载入 CJKfntef 或 xeCJKfntef
宏包。..... 43

v2.5.1 (2020/05/02)
General: zhconv 更名为 ctex-zhconv。..... 1

代码索引

意大利体的数字表示描述对应索引项的页码; 带下划线的数字表示定义对应索引项的代码行号; 罗马字体的数字表示使用对应索引项的代码行号。

Symbols

_ 524, 3271, 3278, 3294, 3297
#\ 490, 3167, 3226
.../afterindent 23
.../aftername 20
.../aftername+ 20
.../afterskip 23
.../afterskip 23
.../aftertitle 20
.../aftertitle+ 20
.../beforeskip 22
.../break 23
.../break+ 23
.../fixskip 23
.../format 18
.../format+ 18
.../hang 22
.../indent 22
.../name 16
.../nameformat 19
.../nameformat+ 19
.../number 17
.../numberformat 19
.../numberformat+ 19
.../titleformat 20
.../titleformat+ 20
.../tocline 24
\/ 1668
\\ 15, 33, 34, 118,
164, 166, 213, 222, 674, 1163, 1245, 2282, 3653, 3654,
3767, 3768, 4016, 4018, 4047, 5132, 5133, 5134, 5168, 5230
\{ 1174, 1175, 5134, 5169
\} 1174, 1175, 5134, 5169
10pt 8, 344
11pt 8, 344
12pt 8, 344

A

\abovedisplayshortskip . 3858, 3866, 3877, 3897, 3905, 3916
\abovedisplayskip 3857, 3860,
3865, 3872, 3876, 3883, 3896, 3899, 3904, 3911, 3915, 3922
\abstractname 2266
abstractname 12
\addCJKfontfeature 1245, 1289
\addCJKfontfeatures 1262

\addcontentsline 2494, 2539, 2543,
2548, 2580, 2584, 2589, 2689, 2694, 2700, 2706, 2835, 2849
\AddEnumerateCounter 2221, 2223, 2224, 2225
\addpenalty 2792, 3494, 3527, 3561, 3575, 3589
\addtocontents 2482, 2488, 2711, 2712
\addvspace 2483, 2489, 2500, 2503,
2711, 2712, 2792, 2796, 3310, 3321, 3329, 3340, 3348, 3359
adobefonts 32
\algorithmname 2273, 2289
algorithmname 13
AlternateFont 29, 34, 1384
\appendix 3004
appendix/name 25, 2993
appendix/number 25, 2993
appendix/numbering 25, 2993
\appendixname 2268, 3638
appendixname 12
\arabic 3541
\AtBeginDocument 579, 1658
\AtBeginDvi 445
\AtBeginShipoutFirst 449, 1814, 1829
\AtBeginUTFCommand 722
\AtEndOfClass 142
\AtEndOfPackage 143
\AtEndUTFCommand 723
autoindent 11, 192, 2075

B

\baselinestretch 2130
\begin 3317, 3336, 3355
\begingroup 418, 2553, 2595, 2615, 2636, 2735,
2760, 2825, 2863, 2896, 3305, 3324, 3343, 4639, 4672, 4679
\belowdisplayshortskip . 3859, 3867, 3878, 3898, 3906, 3917
\belowdisplayskip 3860, 3872, 3883, 3899, 3911, 3922
\bfdefault 1030, 1036, 1374, 1381
\bfseries 2558, 2562, 2600, 2604, 2620,
2641, 2741, 2746, 2765, 3422, 3424, 3428, 3441, 3442,
3445, 3471, 3473, 3479, 3496, 3500, 3521, 3555, 3572, 3586
\bibname 2271, 2274, 2282, 2290, 2296, 2297
bibname 12
bool commands:
\bool_gset_false:N 255, 260
\bool_gset_true:N 250
\bool_if:NTF 83, 1203, 2918, 3108, 3118, 3129, 3657
\bool_lazy_or:nnTF 3993, 4624, 5258

- \bool_new:N 138, 1240
- \bool_set_false:N 1215
- \bool_set_true:N 1233
- \c_false_bool 2952, 2954
- \c_true_bool 2951, 2953
- box commands:
 - \box_ht:N 3938
 - \box_new:N 62
 - \box_use_drop:N 2928
 - \box_wd:N 1862, 2927, 3222
- C**
- c5size 32
- cap 32, 303
- captiondelimiter 33, 3945
- \catcode 4639, 4640, 4641, 4642, 4646, 4647
- CCT 32
- CCTfont 32
- \ccwd 22, 27, 136, 196, 1858, 1903, 1904, 1905, 2081, 2119, 2138, 2139, 2146, 2173, 2174, 2175, 2178, 2180, 2181, 2192, 2203, 2205, 3955, 3956, 3957
- \cdp 1989
- \centering 2596, 2637, 3306, 3325, 3344, 3428, 3440, 3445, 3449, 3479, 3500, 3508, 3543
- \chapter 2666, 3663
- chapter/beforeskip 34
- chapter/fixbeforeskip 34
- chapter/lofskip 24, 2409
- chapter/lotskip 24, 2409
- chapter/numbering 16
- chapter/pagestyle 21, 2409
- \chaptermark 2710
- \chaptername 3468
- char commands:
 - \char_set_catcode:nn 86, 528, 529, 552, 553
 - \char_set_catcode_comment:n 523
 - \char_set_catcode_escape:n 524
 - \char_set_catcode_letter:n 80, 1661, 3242, 5222
 - \char_set_catcode_math_superscript:n 547
 - \char_set_catcode_other:N 490, 942, 3167, 3226
 - \char_set_catcode_other:n 546, 5246
 - \char_set_catcode_space:n 735, 1994
 - \char_value_catcode:n 86, 528, 529, 552, 553
- CharRange 29, 1384
- \Chinese 33, 2217
- \chinese 28, 2215, 2225, 3415, 3478
- \cht 1989
- \CJKbold 4093, 4094, 4104, 4105, 4123, 4124, 4126, 4127, 4129, 4130, 4134, 4135, 4137, 4138, 4140, 4141, 4157, 4158, 4160, 4161, 4163, 4164, 4168, 4169, 4171, 4172, 4174, 4175
- \CJKfamily 537, 538, 539, 540, 1102, 1122, 1329, 1330, 1331, 1332, 1479, 1685, 4614, 4615, 4616, 4617, 4619, 4620, 4622, 4632, 4633
- \CJKfamilydefault 532, 533, 540, 594, 1324, 1325, 1332, 1338, 1340, 1344, 1351, 1364, 1676, 1677, 1681, 1762, 1766, 1774, 1787, 1789
- \CJKfontspec 1262
- \CJKglue 1861, 1875, 1889, 1901
- \CJKhook 607
- \CJKnormal 4086, 4087, 4088, 4089, 4090, 4091, 4092, 4097, 4098, 4099, 4100, 4101, 4102, 4103, 4122, 4125, 4128, 4133, 4136, 4139, 4156, 4159, 4162, 4167, 4170, 4173
- \CJKpunctmapfamily 640, 641, 642, 643, 644, 645, 649, 650, 651, 652, 653, 654
- \CJKrmdefault 513, 533, 534, 537, 1170, 1181, 1293, 1325, 1326, 1329, 1340, 1343, 1344, 1677, 1678, 1682, 1770, 4210, 4213, 4214, 4219, 4220, 4221, 4274, 4277, 4279, 4284, 4285, 4286, 4340, 4343, 4344, 4351, 4352, 4353, 4420, 4423, 4424, 4431, 4432, 4433, 4500, 4502, 4504, 4509, 4510, 4511, 4562, 4563, 4564, 4572, 4573, 4574, 4577
- \CJKsfdefault 514, 535, 538, 1171, 1182, 1300, 1327, 1330, 1679, 1683, 1771, 4211, 4275, 4278, 4341, 4345, 4421, 4425, 4501, 4503, 4565, 4566, 4578
- \CJKtilde 595
- \CJKttdefault 515, 536, 539, 1172, 1183, 1306, 1328, 1331, 1680, 1684, 1772, 4212, 4276, 4342, 4422, 4567, 4579
- clearalternatefont 30, 1563
- \cleardoublepage 2514, 2667, 3438, 3466
- \clearpage 2516, 2667, 3438, 3466
- clist commands:
 - \clist_clear:N 1051, 4952
 - \clist_concat:NNN 1071
 - \clist_const:Nn 620
 - \clist_gput_right:Nn 360, 372, 393, 394
 - \clist_gset:Nn 1316
 - \clist_if_empty:NTF 1409
 - \clist_map_break:n 3087, 3648
 - \clist_map_function:nN .. 1392, 1566, 1568, 1601, 1615
 - \clist_map_inline:Nn 610, 1509
 - \clist_map_inline:nn 345, 632, 1608, 1853, 3082, 3645, 3785, 3820, 4941
 - \clist_new:N 145, 1068, 1317, 1405, 1452, 4970, 5127
 - \clist_put_left:Nn 1072, 3714, 3740
 - \clist_put_right:Nn 1232, 1498, 4964
 - \clist_set:Nn 1055, 1395, 1435, 1436, 1655, 5128
 - \clist_use:Nn 5137
- \clubpenalty 2895, 2902
- \contentsname 2261
- contentsname 12
- continuation 13
- cs commands:
 - \cs_generate_variant:Nn 107, 466, 478, 1017, 1134, 1221, 1241, 1242, 1453, 1469, 1541, 1562, 1601, 1609, 2930, 5184, 5313, 5420
 - \cs_gset_eq:NN 251, 256, 261, 450, 498, 848, 1208, 1211, 3040, 3041, 3046, 3071
 - \cs_gset_nopar:Npx 843
 - \cs_gset_protected:Npn 3005, 3052, 3067
 - \cs_gset_protected:Npx 1099, 1473, 1559, 2981

- \cs_gset_protected_nopar:Npx 885
- \cs_if_eq:NNTF .. 976, 3038, 3070, 5195, 5201, 5346, 5426
- \cs_if_exist:NTF 44, 336, 475, 837, 853, 883,
917, 995, 1472, 1557, 1953, 3153, 3370, 3385, 3663, 3665
- \cs_if_exist_p:N 5259, 5260
- \cs_if_exist_use:N 1118, 1119
- \cs_if_exist_use:NTF 4, 820, 1140, 1577, 5191
- \cs_if_free:NTF 460, 470, 612, 627, 1205, 1373,
1920, 1923, 2094, 2108, 2221, 3101, 3146, 3192, 3197, 3201
- \cs_new:Npn 128, 135, 902, 932, 937, 939,
943, 1047, 1113, 1121, 1166, 1177, 1630, 2215, 2391,
2955, 2963, 2971, 2990, 3126, 3190, 3195, 4971, 5303, 5421
- \cs_new:Npx 3964, 3970
- \cs_new_eq:NN
... 6, 7, 8, 9, 20, 23, 122, 123, 124, 140, 141, 337, 467,
598, 658, 862, 887, 953, 983, 1289, 1296, 1783, 1901,
1911, 1917, 2216, 2217, 2218, 2449, 2468, 2471, 2939,
2949, 2950, 2951, 2952, 2953, 2954, 3004, 3028, 3029,
3388, 3661, 3666, 4628, 4629, 4946, 5249, 5250, 5350, 5422
- \cs_new_protected:Npn 65, 68, 70, 72, 75, 90, 93, 126,
142, 143, 151, 153, 158, 440, 442, 444, 458, 468, 473,
479, 511, 519, 542, 559, 599, 605, 608, 625, 630, 638,
647, 656, 724, 730, 817, 827, 840, 845, 851, 904, 915,
927, 946, 955, 971, 985, 1010, 1048, 1069, 1078, 1097,
1114, 1124, 1149, 1187, 1199, 1201, 1223, 1225, 1248,
1334, 1359, 1368, 1391, 1393, 1400, 1407, 1413, 1420,
1422, 1427, 1433, 1454, 1463, 1470, 1476, 1487, 1495,
1501, 1507, 1525, 1534, 1542, 1554, 1572, 1589, 1607,
1610, 1612, 1625, 1644, 1652, 1666, 1669, 1671, 1706,
1720, 1733, 1746, 1760, 1794, 1798, 1836, 1844, 1848,
1858, 1872, 1902, 1906, 1918, 1968, 2093, 2095, 2120,
2155, 2165, 2170, 2324, 2354, 2402, 2432, 2439, 2444,
2452, 2459, 2466, 2472, 2474, 2477, 2493, 2907, 2913,
2915, 2923, 2931, 2940, 3021, 3031, 3063, 3074, 3094,
3099, 3104, 3114, 3124, 3168, 3216, 3234, 3366, 3375,
3380, 3758, 3773, 3805, 3810, 3840, 3846, 3960, 3976,
3986, 3988, 4938, 4947, 4962, 5142, 5144, 5146, 5148,
5158, 5164, 5171, 5177, 5179, 5185, 5190, 5193, 5199,
5206, 5211, 5216, 5227, 5231, 5236, 5241, 5251, 5256,
5275, 5326, 5341, 5351, 5363, 5372, 5391, 5396, 5408, 5423
- \cs_new_protected:Npx 483, 1272, 3155
- \cs_new_protected_nopar:Npn 863, 877
- \cs_replacement_spec:N 5267, 5289
- \cs_set:Npn 1914, 3207
- \cs_set_eq:NN 125, 197, 202, 207, 557, 577,
602, 606, 614, 628, 856, 857, 954, 984, 1002, 1045, 1102,
1472, 1479, 1489, 1490, 1491, 1557, 1558, 1668, 1765,
1932, 1935, 1936, 1939, 2232, 2237, 2240, 2245, 2473,
2475, 2942, 2943, 2944, 2945, 2946, 2947, 3035, 3078, 3088
- \cs_set_nopar:Npn 1705
- \cs_set_nopar:Npx 3116, 5150
- \cs_set_protected:Npn
..... 601, 988, 1003, 1875, 3148, 3205, 3847, 5254, 5318
- \cs_set_protected:Npx 1908, 5243, 5244
- \cs_to_str:N 58, 1271, 5252
- \cs_undefine:N . 1082, 1083, 1087, 1088, 1492, 1940, 5191
- cs4size 32
- \csname 419, 422, 423,
2793, 2802, 2803, 2804, 2811, 2831, 2832, 2834, 2837,
2846, 2847, 2848, 2851, 4671, 4673, 4674, 4675, 4680, 4684
- \CTeX 28, 3941
- ctex commands:
- \ctex_add_cmap:n 463, 468
- \ctex_add_cmap:Nn 471, 473, 478
- \ctex_add_dvi_zhmap: 444, 446, 450
- \ctex_add_to_selectfont:n 1906, 1959, 1987, 2014, 2107
- \ctex_after_end_preamble:n 39, 125, 446, 5142
- \ctex_appto_cmd:NnnTF 494, 1660, 5236
- \ctex_assign_heading_name:nn 2360, 2399, 2997
- \ctex_at_begin_package:nn 138, 5158
- \ctex_at_end:n .. 142, 274, 312, 318, 329, 334, 1901, 4054
- \ctex_at_end_package:nn 138, 447, 497, 732,
993, 1808, 1810, 1828, 1833, 1840, 2219, 3050, 3065,
3134, 3141, 3200, 3202, 3203, 3239, 3282, 3394, 3943, 5171
- \ctex_at_end_preamble:n 339, 584, 591, 1333, 1693, 5142
- \ctex_auto_ignorespaces: 598, 606, 2024
- \ctex_CJK_input:n 542
- \ctex_declare_math_sizes:nnnn 3805, 3839
- \ctex_define:n 64, 1563, 1602, 2016, 2037, 2075, 2109,
2233, 2259, 2287, 2297, 2300, 2303, 2431, 2993, 3945, 4021
- \ctex_define_option:n 64, 173, 366, 369
- \ctex_deprecated_command:Nn
..... 151, 2058, 2060, 2068, 2070, 2202, 2209
- \ctex_deprecated_option:n
..... 151, 155, 211, 220, 311, 317, 321, 328, 333, 3949
- \ctex_deprecated_option:nn 151, 278
- \ctex_detect_platform: 3976, 3991, 4006
- \ctex_family_cmap:nn 458, 467
- \ctex_file_input:n
... 90, 375, 659, 2050, 3672, 3707, 3712, 3717, 3718,
3733, 3738, 3744, 3745, 3850, 3851, 4013, 4054, 4391, 4392
- \ctex_file_wrapper:nnn 74, 91, 95, 521, 544
- \ctex_fix_varioref_label:n 3378, 3380
- \ctex_fontset_case:nnn 3964, 4325, 4544
- \ctex_fontset_case:nnnn . 3970, 4186, 4250, 4395, 4483
- \ctex_fontset_error:n 3960, 4187,
4223, 4251, 4288, 4355, 4396, 4398, 4399, 4435, 4484, 4513
- \ctex_gadd_hook:Nn 607, 5177
- \ctex_gadd_package_hook:nnn 5162, 5175, 5177
- \ctex_get_macro_meaning:N 5321
- \ctex_get_macro_meaning:NTF 5317, 5343, 5399
- \ctex_hang_from:n 2915
- \ctex_heading_glue:n 2931
- \ctex_heading_hang:Nnn 2909, 2914, 2915
- \ctex_hypersetup:n
..... 75, 337, 1791, 1803, 1806, 1821, 1824, 1827
- \ctex_if_autoindent_touched:TF 140, 197, 202, 207, 3699
- \ctex_if_ccglue_touched:TF 1886, 2126
- \ctex_if_ccglue_touched_p: 1886
- \ctex_if_platform_macos:TF 3981, 3986
- \ctex_if_preamble:TF 124, 4025
- \ctex_if_rescanable:NnTF 5341, 5375

- \ctex_ignorespaces_case:N [599](#), [2021](#), [2024](#), [2027](#)
- \ctex_indent_box:n [2439](#)
- \ctex_load_fontset: [3988](#), [4033](#), [4053](#)
- \ctex_load_zhmap:nnnn
..... [511](#), [4217](#), [4282](#), [4349](#), [4429](#), [4507](#), [4570](#)
- \ctex_ltj_add_font_features:n [1223](#), [1286](#)
- \ctex_ltj_add_font_features:nn [1223](#), [1224](#)
- \ctex_ltj_add_kyenc:n [744](#), [1026](#)
- \ctex_ltj_char_range_key:nn [1649](#), [1652](#)
- \ctex_ltj_clear_alt_font:n [781](#), [1520](#)
- \ctex_ltj_clear_alternate_font:n [1566](#), [1572](#)
- \ctex_ltj_declare_alternate_shape:nnnnnn [1481](#), [1501](#)
- \ctex_ltj_declare_char_range:n [1604](#), [1607](#), [1609](#)
- \ctex_ltj_declare_char_range:nn [1611](#), [1612](#)
- \ctex_ltj_def_char_range_key:n [1617](#), [1644](#)
- \ctex_ltj_ensure_default_family: [1334](#), [1779](#)
- \ctex_ltj_extract_font: [856](#), [863](#)
- \ctex_ltj_family_if_exist:nNTF
..... [1126](#), [1135](#), [1338](#), [1343](#), [1361](#), [1364](#), [1575](#)
- \l_ctex_ltj_family_tl
[1128](#), [1133](#), [1190](#), [1191](#), [1206](#), [1210](#), [1213](#), [1224](#), [1569](#), [1570](#)
- \ctex_ltj_fontspec:nn [1187](#), [1199](#), [1234](#), [1281](#)
- \ctex_ltj_get_and_define_fonts:nN [946](#)
- \ctex_ltj_get_and_define_fonts_al:nN [950](#), [953](#)
- \ctex_ltj_get_and_define_fonts_ja:nN [949](#), [955](#)
- \ctex_ltj_if_alternate_shape_exist:nTF
..... [866](#), [879](#), [897](#), [906](#)
- \ctex_ltj_if_jfont:nTF [60](#), [932](#), [944](#), [948](#)
- \ctex_ltj_if_jfont_math:NTF [939](#), [978](#)
- \ctex_ltj_is_kenc:n [747](#), [934](#)
- \ctex_ltj_math_group_hook: [59](#), [59](#), [977](#), [983](#), [1003](#)
- \ctex_ltj_patch_external_font:n [750](#), [903](#)
- \ctex_ltj_pickup_alt_font:nn [787](#), [908](#)
- \ctex_ltj_pickup_font: ... [833](#), [851](#), [922](#), [958](#), [960](#), [962](#)
- \ctex_ltj_reset_alternate_font:n [1568](#), [1589](#)
- \ctex_ltj_save_alternate_family:Nnnn [1446](#), [1467](#), [1470](#)
- \ctex_ltj_save_alternate_seq:Nn [1417](#), [1422](#)
- \ctex_ltj_save_char_range:n [1615](#), [1622](#)
- \ctex_ltj_save_char_range:nn [1624](#), [1625](#)
- \ctex_ltj_select_alternate_font: [904](#), [1962](#)
- \ctex_ltj_select_font: [817](#), [1961](#)
- \ctex_ltj_set_alt_font:nnnn [772](#), [1536](#), [1539](#)
- \ctex_ltj_set_alternate_family:nn [1421](#), [1454](#)
- \ctex_ltj_set_alternate_family:nnn . [1466](#), [1474](#), [1476](#)
- \ctex_ltj_set_alternate_family:Nnnnn [1458](#), [1463](#), [1469](#)
- \ctex_ltj_set_alternate_family:nnTF [1056](#), [1407](#)
- \ctex_ltj_set_alternate_prop:n [1386](#), [1391](#)
- \ctex_ltj_set_alternate_shape:n [1527](#), [1534](#), [1541](#), [1546](#)
- \ctex_ltj_set_alternate_shape:nnN [1513](#), [1542](#)
- \ctex_ltj_set_alternate_shape:nnn [1516](#), [1522](#)
- \ctex_ltj_set_alternate_shape:nnnn [1524](#), [1525](#)
- \ctex_ltj_set_alternate_shape:Nnnnnnn .. [1504](#), [1507](#)
- \ctex_ltj_set_family:nnn
..... [1048](#), [1218](#), [1265](#), [1275](#), [1293](#), [1300](#), [1306](#), [1312](#)
- \ctex_ltj_set_kanjiskip:N [1669](#), [1880](#)
- \ctex_ltj_set_math_letter:NN [985](#), [997](#), [1005](#)
- \ctex_ltj_set_xkanjiskip:N [1669](#), [1975](#)
- \ctex_ltj_subst_font: [857](#), [877](#)
- \ctex_ltj_swap_cs:NN [1480](#), [1487](#)
- \ctex_ltj_switch_family:n
..... [1123](#), [1124](#), [1134](#), [1191](#), [1219](#), [1273](#)
- \ctex_ltj_update_mathfont: [1356](#), [1359](#)
- \ctex_ltj_update_mathfont:n [1362](#), [1365](#), [1368](#)
- \ctex_ltj_use_jfont: [767](#), [873](#)
- \ctex_ltj_use_math_group:Nn [971](#), [1002](#)
- \ctex_ltj_zero_globaldefs: [727](#), [810](#)
- \ctex_make_spa:nn [4938](#)
- \ctex_mono_jfm:n [1010](#), [1018](#), [2044](#)
- \ctex_parse_name:NN [1943](#), [5247](#), [5251](#)
- \ctex_parse_name:NNn [5252](#), [5256](#), [5313](#)
- \ctex_patch_boot:NnnnTF
..... [5208](#), [5213](#), [5218](#), [5233](#), [5238](#), [5241](#)
- \ctex_patch_cmd:Nnn
..... [1701](#), [3170](#), [3176](#), [3182](#), [3254](#), [3257](#), [3261](#),
[3267](#), [3277](#), [3285](#), [3288](#), [3293](#), [3296](#), [3955](#), [3956](#), [3957](#), [5216](#)
- \ctex_patch_cmd_all:NnnnTF [5211](#)
- \ctex_patch_cmd_once:NnnnTF [1695](#), [1945](#), [5206](#)
- \ctex_patch_failure:N
..... [500](#), [1664](#), [1950](#), [3233](#), [3245](#), [5225](#), [5227](#)
- \ctex_plane_to_utfxvibe:Nn [558](#)
- \ctex_preto_cmd:NnnTF [3230](#), [3241](#), [5231](#)
- \ctex_punct_map_bfseries:nn [630](#),
[4213](#), [4220](#), [4277](#), [4278](#), [4285](#), [4344](#), [4345](#), [4346](#), [4352](#),
[4424](#), [4425](#), [4426](#), [4432](#), [4502](#), [4503](#), [4510](#), [4563](#), [4566](#), [4573](#)
- \ctex_punct_map_family:nn
..... [625](#), [4210](#), [4211](#), [4212](#), [4219](#),
[4274](#), [4275](#), [4276](#), [4284](#), [4340](#), [4341](#), [4342](#), [4351](#), [4420](#),
[4421](#), [4422](#), [4431](#), [4500](#), [4501](#), [4509](#), [4562](#), [4565](#), [4567](#), [4572](#)
- \ctex_punct_map_itshape:nn [647](#), [4214](#), [4221](#),
[4279](#), [4286](#), [4343](#), [4353](#), [4423](#), [4433](#), [4504](#), [4511](#), [4564](#), [4574](#)
- \ctex_punct_map_series:nnn [634](#), [635](#), [638](#)
- \ctex_punct_set:n [608](#), [4209](#), [4218](#),
[4273](#), [4283](#), [4339](#), [4350](#), [4419](#), [4430](#), [4499](#), [4508](#), [4548](#), [4571](#)
- \ctex_punct_space:nn [656](#)
- \ctex_scheme_input:n [93](#), [3396](#), [3674](#)
- \ctex_select_size: .. [2082](#), [2089](#), [2092](#), [2093](#), [2114](#), [2168](#)
- \ctex_set:n [64](#), [275](#), [2052](#), [2072](#), [3698](#), [3700](#)
- \ctex_set:nn
..... [31](#), [64](#), [156](#), [2062](#), [3399](#), [3457](#), [3485](#), [3517](#), [3552](#),
[3569](#), [3583](#), [3599](#), [3607](#), [3615](#), [3623](#), [3630](#), [3631](#), [3632](#), [4061](#)
- \ctex_set_default_ccwd:Nn [126](#), [206](#), [2088](#), [2113](#)
- \ctex_set_deprecated_option:n
..... [151](#), [185](#), [186](#), [240](#), [241](#), [242](#), [264](#), [269](#), [303](#), [304](#)
- \ctex_set_font_size:Nnn [3840](#), [3855](#),
[3863](#), [3874](#), [3885](#), [3886](#), [3887](#), [3888](#), [3889](#), [3890](#), [3891](#),
[3894](#), [3902](#), [3913](#), [3924](#), [3925](#), [3926](#), [3927](#), [3928](#), [3929](#), [3930](#)
- \ctex_set_ignorespaces: [599](#), [607](#)
- \ctex_set_upfamily:nnn ... [1706](#), [1753](#), [4233](#), [4234](#),
[4235](#), [4236](#), [4298](#), [4299](#), [4300](#), [4301](#), [4365](#), [4366](#), [4367](#),
[4368](#), [4369](#), [4370](#), [4444](#), [4445](#), [4446](#), [4447](#), [4448](#), [4449](#),
[4523](#), [4524](#), [4525](#), [4590](#), [4591](#), [4592](#), [4593](#), [4594](#), [4595](#), [4596](#)

- \ctex_set_upfonts:nnnnnn [1746](#), [1756](#), [4226](#), [4291](#), [4358](#), [4583](#)
- \ctex_set_upmap:nnn .. [1720](#), [1748](#), [1749](#), [1750](#), [1751](#),
[1754](#), [4371](#), [4372](#), [4438](#), [4439](#), [4440](#), [4441](#), [4522](#), [4597](#), [4598](#)
- \ctex_set_upmap_unicode:nnn [1733](#), [1755](#), [4442](#), [4443](#), [4516](#), [4518](#), [4520](#)
- \ctex_set_zhmap:n [439](#), [516](#), [1722](#), [1735](#)
- \ctex_titleps_hook: [3166](#), [3201](#), [3202](#)
- \ctex_titlesec_hook: [3074](#), [3138](#)
- \ctex_update_ccglue: [1872](#), [2133](#), [2152](#), [2196](#)
- \ctex_update_ccwd: [1858](#), [2127](#), [2204](#)
- \ctex_update_default_family: [49](#), [584](#), [1333](#), [1693](#), [1759](#)
- \ctex_update_em_unit: [1902](#), [2122](#), [2172](#)
- \ctex_update_kanjisize: [1989](#)
- \ctex_update_parindent: [2100](#), [2155](#), [2197](#)
- \ctex_update_size: [2094](#), [2095](#)
- \ctex_update_stretch: [2099](#), [2120](#)
- \ctex_update_xkanjiskip: [1967](#), [1987](#)
- \ctex_update_ziju: [2102](#), [2170](#)
- \ctex_use_package_hook:nn [5185](#), [5196](#), [5202](#)
- \ctex_use_zhmap: [442](#), [445](#), [449](#)
- \ctex_varioref_hook: [3375](#), [3386](#), [3394](#)
- \ctex_zhmap_case:nnn
.. [141](#), [251](#), [256](#), [261](#), [4189](#), [4253](#), [4327](#), [4403](#), [4486](#), [4546](#)
- \ctex_zhmap_input:n [516](#), [519](#)
- \ctex_zihao:n [3757](#), [3758](#)
- \ctex_ziju:n [2164](#), [2165](#)
- ctex internal commands:
- \g__ctex_after_end_preamble_hook_tl [5145](#), [5149](#), [5157](#)
- __ctex_assign_heading_name:nnn [2399](#)
- \l__ctex_autoindent_tl
.. [139](#), [196](#), [201](#), [206](#), [2081](#), [2085](#), [2088](#), [2157](#), [2160](#), [2435](#)
- __ctex_calc_bounds:nn [4966](#), [4967](#), [4971](#)
- __ctex_ccglue: [1889](#), [1901](#)
- \l__ctex_ccglue_skip [1876](#),
[1880](#), [1883](#), [1885](#), [1893](#), [1897](#), [2131](#), [2142](#), [2151](#), [2187](#), [2195](#)
- \c__ctex_class_tl
. [98](#), [100](#), [397](#), [402](#), [407](#), [412](#), [3661](#), [3667](#), [3669](#), [3670](#), [3672](#)
- __ctex_clear_fntef_color:n ... [1836](#), [1844](#), [1848](#), [1855](#)
- \c__ctex_cmap_encoding_seq [456](#), [462](#)
- \l__ctex_ctexcap_options_clist [5127](#), [5128](#), [5137](#)
- __ctex_def_heading_keys:n [2354](#), [2429](#)
- __ctex_default_ccwd_aux:n [127](#), [128](#)
- __ctex_default_ccwd_aux:w [131](#), [135](#)
- \g__ctex_encoding_tl [137](#), [230](#), [233](#), [236](#),
[383](#), [387](#), [504](#), [507](#), [586](#), [594](#), [1804](#), [2213](#), [3702](#), [3729](#), [3743](#)
- \g__ctex_end_preamble_hook_tl [5143](#), [5147](#), [5156](#)
- \c__ctex_engine_file_str [43](#), [2050](#)
- \c__ctex_engine_str [43](#)
- \l__ctex_family_default_init_tl [1762](#), [1782](#)
- __ctex_family_default_wrap:n [1765](#), [1783](#), [1786](#)
- __ctex_fix_varioref_label:n [3380](#)
- \g__ctex_font_size_int [168](#),
[179](#), [180](#), [181](#), [355](#), [359](#), [391](#), [3719](#), [3720](#), [3746](#), [3747](#), [3848](#)
- \c__ctex_font_size_prop [3760](#), [3771](#), [3815](#), [3842](#)
- \c__ctex_font_size_seq [3769](#), [3772](#), [3783](#)
- \l__ctex_font_size_tl [3760](#), [3761](#), [3807](#), [3808](#), [3842](#), [3843](#)
- \g__ctex_fontset_tl [239](#), [3979](#), [3982](#), [3983](#), [3990](#), [3994](#),
[3995](#), [3998](#), [3999](#), [4002](#), [4008](#), [4013](#), [4030](#), [4032](#), [4037](#), [4052](#)
- __ctex_get_font_sizes:Nn [3807](#), [3810](#)
- __ctex_get_macro_meaning:w [5317](#)
- \l__ctex_heading_bool [280](#), [3657](#)
- \l__ctex_heading_skip [2456](#), [2463](#), [2465](#), [2467](#),
[2468](#), [2479](#), [2480](#), [2483](#), [2485](#), [2486](#), [2489](#), [2934](#), [2935](#), [2936](#)
- \c__ctex_headings_cs_seq [3023](#), [3036](#), [3045](#), [3068](#)
- \c__ctex_headings_seq ... [2314](#), [2426](#), [3151](#), [3214](#), [3377](#)
- __ctex_hookto_cmd:Nnnnw [5233](#), [5238](#), [5396](#)
- __ctex_hookto_cmd_parameter:Nnnnw [5403](#), [5423](#)
- __ctex_hookto_cmd_parameterless:Nnnnw . [5402](#), [5408](#)
- __ctex_initial_heading:n [2324](#), [2428](#)
- __ctex_insert_indent: [2442](#), [2444](#), [2921](#)
- __ctex_left_hook_aux:nn [5421](#)
- \l__ctex_line_spread_fp
..... [115](#), [189](#), [3721](#), [3722](#), [3748](#), [3749](#), [3934](#), [3936](#)
- \l__ctex_line_stretch_tl [2113](#), [2118](#), [2119](#), [2123](#)
- \l__ctex_ltj_add_alternate_bool [1203](#), [1215](#), [1233](#), [1240](#)
- __ctex_ltj_alternate_cs:n
..... [1083](#), [1086](#), [1087](#), [1088](#), [1118](#), [1119](#), [1121](#), [1206](#),
[1209](#), [1210](#), [1212](#), [1213](#), [1417](#), [1459](#), [1519](#), [1538](#), [1578](#), [1595](#)
- \l__ctex_ltj_alternate_family_tl [1483](#), [1486](#)
- \l__ctex_ltj_alternate_options_clist
..... [1436](#), [1437](#), [1448](#), [1452](#)
- \l__ctex_ltj_alternate_prop
..... [1052](#), [1402](#), [1403](#), [1406](#), [1416](#), [1424](#)
- \l__ctex_ltj_base_CJKfamily_tl
..... [1053](#), [1067](#), [1116](#), [1519](#), [1538](#)
- \l__ctex_ltj_base_family_tl [69](#),
[1084](#), [1111](#), [1117](#), [1465](#), [1505](#), [1558](#), [1575](#), [1581](#), [1582](#), [1594](#)
- \l__ctex_ltj_char_range_clist
... [1051](#), [1388](#), [1409](#), [1435](#), [1447](#), [1460](#), [1482](#), [1504](#), [1655](#)
- \g__ctex_ltj_char_range_prop [1511](#), [1612](#)
- \l__ctex_ltj_char_range_tl
..... [1511](#), [1514](#), [1614](#), [1616](#), [1618](#), [1620](#), [1627](#)
- __ctex_ltj_check_family:n [1078](#), [1415](#)
- \l__ctex_ltj_current_font_tl
..... [55](#), [820](#), [824](#), [825](#), [837](#), [883](#), [888](#)
- \l__ctex_ltj_current_shape_tl [906](#), [909](#), [912](#), [913](#)
- __ctex_ltj_declare_char_range:nn [1608](#), [1610](#)
- \g__ctex_ltj_default_features_clist [1071](#), [1316](#), [1317](#)
- __ctex_ltj_family_csname:n ... [1082](#), [1099](#), [1113](#), [1140](#)
- \g__ctex_ltj_family_font_name_prop
..... [1058](#), [1075](#), [1080](#), [1151](#), [1227](#), [1336](#), [1348](#)
- \g__ctex_ltj_family_font_options_prop
..... [1059](#), [1075](#), [1230](#)
- \g__ctex_ltj_family_int . [1193](#), [1195](#), [1497](#), [1498](#), [1500](#)
- \g__ctex_ltj_family_name_prop
..... [1075](#), [1084](#), [1105](#), [1137](#), [1465](#)
- __ctex_ltj_family_unknown_warning:n [1131](#), [1149](#), [1586](#)
- \l__ctex_ltj_font_options_clist [1055](#), [1060](#),
[1061](#), [1062](#), [1068](#), [1104](#), [1231](#), [1232](#), [1235](#), [1456](#), [1457](#), [1461](#)
- \g__ctex_ltj_fontname_seq [842](#), [847](#), [850](#)
- __ctex_ltj_fontspec:nnn [1194](#), [1201](#), [1221](#)

- \g_ctex_ltj_fontspec_family_tl 1103, 1106, 1112, 1117, 1142
- \g_ctex_ltj_fontspec_prop 1189, 1217, 1222
- _ctex_ltj_gset_family_cs:nn 1063, 1097
- _ctex_ltj_if_alt_set:nTF 802, 899
- _ctex_ltj_if_jfont_math:w 940, 943
- \l_ctex_ltj_jfm_tl 1010, 1029, 1031, 1073
- _ctex_ltj_ltj_if_jfont:w 934, 937
- \c_ctex_ltj_math_fam_int 1037, 1038
- \c_ctex_ltj_math_family_tl 1370, 1372, 1374, 1377, 1381
- \c_ctex_ltj_math_tl 1032, 1033, 1035, 1037, 1312, 1361, 1371, 1376, 1380
- _ctex_ltj_msg_def_family_map:n 1164, 1166
- _ctex_ltj_msg_family_map:n 1096, 1163, 1177
- _ctex_ltj_noembed_wrap:n 1045, 1047
- _ctex_ltj_pass_args:nnnn 1248, 1264, 1274, 1280, 1292, 1299, 1305, 1311
- _ctex_ltj_patch_external_font:w 869, 902
- _ctex_ltj_pickup_alt_font:Nn 795, 930
- _ctex_ltj_pop_fontname: 836, 845, 924
- _ctex_ltj_post_arg:w 1251, 1257
- _ctex_ltj_push_alternate_prop:n 1392, 1393
- _ctex_ltj_push_alternate_prop:nn . 1397, 1400, 1656
- _ctex_ltj_push_fontname:n 832, 840, 921
- \c_ctex_ltj_range_max_int 1639, 1643
- \c_ctex_ltj_range_min_int 1638, 1642
- _ctex_ltj_range_normalization:nn . 1529, 1628, 1630
- \g_ctex_ltj_reset_alternate_prop 1089, 1580, 1592, 1600
- _ctex_ltj_save_alternate_auxi:w 1425, 1427
- _ctex_ltj_save_alternate_auxii:w . 1430, 1431, 1433
- _ctex_ltj_save_alternate_shape:Nn 1518, 1537, 1554
- _ctex_ltj_select_font_aux: 821, 827
- _ctex_ltj_set_alternate_family:n 1107, 1114
- _ctex_ltj_set_alternate_family_aux:nnn 1411, 1420
- _ctex_ltj_set_family_aux:nnn 1410, 1413
- _ctex_ltj_tmp:w 1489, 1491, 1492
- \l_ctex_ltj_tmp_clist 1395, 1396, 1398, 1405
- \l_ctex_ltj_tmp_tl 847, 848, 1054, 1055, 1080, 1091, 1094, 1228, 1236, 1270, 1273, 1275, 1338, 1343, 1361, 1362, 1364, 1365, 1439, 1441, 1442, 1449
- \g_ctex_ltj_unknown_family_seq ... 1153, 1155, 1160
- _ctex_ltj_update_family_uid:N ... 1061, 1456, 1494
- _ctex_ltj_use_global_options:N 1062, 1069, 1437, 1457
- \c_ctex_msyh_suffix_tl 4536, 4551, 4552, 4558, 4559, 4587, 4588
- _ctex_package_loaded_warning:nn 5161, 5164
- \l_ctex_parameter_str 5314, 5333, 5354, 5401
- \l_ctex_parameter_tl 5354, 5359, 5370
- _ctex_parse_name:nNNNNN 5266, 5275
- _ctex_parse_name:w 5288, 5294, 5303
- _ctex_patch_cmd:Nnnnnw 5208, 5213, 5218, 5372
- _ctex_patch_false:w 5244, 5250, 5387, 5389, 5406, 5432
- _ctex_patch_rebuild:Nn 5345, 5351, 5384, 5425, 5429
- _ctex_patch_replace:nnnTF 5379, 5391
- _ctex_patch_rescan:NNn 5353, 5354, 5355, 5363
- _ctex_patch_toc_width:n 3225
- _ctex_patch_true:w 5243, 5249, 5385, 5418, 5430
- _ctex_plus_key_aux:nn 2381, 2383, 2384, 2385, 2386, 2387, 2388, 2391
- \l_ctex_prefix_str 5314, 5332, 5353, 5410, 5411
- \l_ctex_prefix_tl 5353, 5358, 5369
- \l_ctex_punct_bounds_clist ... 4952, 4958, 4964, 4970
- \c_ctex_punct_family_clist 610, 620
- \l_ctex_punct_font 4950, 4951
- \c_ctex_punct_seq 4930, 4953
- \l_ctex_punct_tl 266, 581, 582, 666, 1018, 2041, 2042, 2043, 2044
- _ctex_rebuild_cmd:w ... 5345, 5346, 5350, 5425, 5426
- \l_ctex_replacement_str 5314, 5334, 5355, 5393, 5394, 5428
- \l_ctex_replacement_tl 5355, 5360, 5371
- _ctex_restore_selectfont: 1918, 1940, 1949
- _ctex_right_hook_aux:nn 5422
- _ctex_save_bounds:n 4954, 4962
- _ctex_save_cmap:Nn 475, 479
- _ctex_save_font_size:nn 3771
- \l_ctex_scheme_tl 291, 3396, 3674
- \g_ctex_section_depth_int 108, 285, 287, 3597, 3613, 3629
- \c_ctex_section_headings_seq . 2310, 2316, 2978, 3092
- _ctex_set_font_size:nnNn 3843, 3846
- \g_ctex_spa_iow 4940, 4943, 4945, 4955
- \c_ctex_std_class_tl 112, 3645, 3660, 3661, 3727, 3750
- \g_ctex_std_options_clist 144, 360, 372, 393, 394, 398, 403, 408, 413
- _ctex_titlesec_format:n 3088, 3096, 3099
- _ctex_titlesec_format:Nn 3078, 3102, 3104
- _ctex_titlesec_format_explicit:Nn ... 3079, 3114
- _ctex_titlesec_hook:n 3088, 3092, 3094
- _ctex_titlesec_spacing:Nn 3097, 3124
- _ctex_titlesec_spacing:nnnnnn 3125, 3126
- _ctex_tmp:N 1914, 1943
- _ctex_tmp:w 5254, 5307, 5318, 5339
- \l_ctex_tmp_box 60, 1861, 1862, 2926, 2927, 2928, 3218, 3222
- \l_ctex_tmp_dim 60, 2123, 2124, 2129, 2132, 2138, 2177, 2179, 2181, 2190, 2192
- \l_ctex_tmp_int 60, 562, 564, 565, 567, 571, 573, 2137, 2139, 2140, 2146, 2147, 2179, 2180, 2181, 2184, 2185, 2190, 2192, 2977, 2980, 2984
- \l_ctex_tmp_tl 60, 344, 351, 366, 367, 481, 482, 561, 563, 1916, 1945, 2356, 2411, 2424, 2431, 3726, 3727, 3815, 3816
- _ctex_use_package_hook_aux:N 5188, 5190
- \c_ctex_version_tl 3, 6, 7, 8, 9, 20, 23
- _ctex_write_family:nn 4942, 4947
- \l_ctex_xkanjiskip_skip 1967
- \l_ctex_xkanjiskip_tl 1974, 1979, 1980
- \c_ctex_zero_tl 2097, 2104, 2106

`\g__ctex_zhmap_tl` 441, 443, 452
`\g__ctex_zhmCJK_bool` 138, 250, 255, 260, 503, 585
`\l__ctex_ziju_dim` 2173, 2174, 2178, 2189, 2191, 2195, 2199
`\l__ctex_ziju_tl` 2097, 2105, 2106, 2167, 2173
`\ctexbibname` 2283, 2303
`\CTEXcounter` 2218
`\CTEXdigits` 28, 2228
`CTEXfilltwosides` 33, 34
`\CTEXifname` 26, 2471, 2561, 2603, 2745, 2957, 2965, 2973,
3209, 3256, 3259, 3263, 3275, 3279, 3287, 3290, 3295, 3298
`\CTEXindent` 33, 2200
`\CTEXnoindent` 33, 2200
`\CTEXnumber` 28, 2228
`\CTEXnumberline` 2971, 2991, 3475
`\CTEXoptions` 33, 2054
`\ctexset` 5, 2051, 2061, 2071
`\CTEXsetfont` 33, 2092
`\CTEXsetup` 33, 2054
`\CTEXsout` 33, 34
`\ctexspadef` 656, 4957
`\CTEXthechapter` 25, 2688, 2966, 3144, 3275, 3295
`\CTEXtheparagraph` 25
`\CTEXthepart` 25, 2958, 3149
`\CTEXthesection` 25, 3256, 3263, 3279, 3287, 3298
`\CTEXthesubparagraph` 25
`\CTEXthesubsection` 25, 3259, 3290
`\CTEXthesubsubsection` 25
`\CTEXunderdblline` 33, 34
`\CTEXunderdot` 33, 34
`\CTEXunderline` 33, 34
`\CTEXunderwave` 33, 34
`\CTEXxout` 33, 34
`\CurrentOption` 372
`\cwd` 1989

D

`declarecharrange` 29, 1602
`\DeclareErrorKanjiFont` 1687
`\DeclareFontEncoding` 1020
`\DeclareFontFamily`
..... 489, 1027, 4072, 4073, 4074, 4075, 4076, 4077
`\DeclareFontShape` 1028, 1030, 1710, 1711, 1714,
1715, 1716, 1717, 4086, 4087, 4088, 4089, 4090, 4091,
4092, 4093, 4094, 4097, 4098, 4099, 4100, 4101, 4102,
4103, 4104, 4105, 4108, 4109, 4110, 4111, 4114, 4115,
4116, 4117, 4122, 4123, 4124, 4125, 4126, 4127, 4128,
4129, 4130, 4133, 4134, 4135, 4136, 4137, 4138, 4139,
4140, 4141, 4144, 4145, 4146, 4149, 4150, 4151, 4156,
4157, 4158, 4159, 4160, 4161, 4162, 4163, 4164, 4167,
4168, 4169, 4170, 4171, 4172, 4173, 4174, 4175, 4178, 4181
`\DeclareFontSubstitution` 1023
`\DeclareKanjiFamily`
..... 1708, 1709, 4078, 4079, 4080, 4081, 4082, 4083
`\DeclareKanjiSubstitution` 1688, 1689
`\DeclareMathSizes` 3808
`\DeclareSymbolFont` 1033, 1371, 1690

`\def` 425, 2534, 2575, 2611, 2632,
2648, 2679, 2720, 2730, 2755, 2775, 2801, 2805, 2842,
2857, 2873, 2874, 2878, 3868, 3879, 3907, 3918, 4645, 4681
`\defaultCJKfontfeatures` 1290
`\defbeamertemplate` 3304, 3323, 3342
`\defbeamertemplatealias` 3361, 3362, 3363
`\detokenize` 426
dim commands:
`\dim_abs:n` 2191
`\dim_add:Nn` 2174
`\dim_compare:nNnTF`
..... 2124, 2159, 2175, 2182, 2446, 2455, 2935
`\dim_eval:n` 2144, 2190
`\dim_max:nn` 3221
`\dim_min:nn` 2191
`\dim_new:N` 63, 1871, 1990, 1991, 1992, 2199
`\dim_set:Nn` 1862, 1903, 1904, 1905, 2123,
2129, 2160, 2173, 2177, 2205, 2441, 2462, 2917, 3219, 3938
`\dim_sub:Nn` 2181
`\dim_to_decimal:n` 3779, 3780, 3817
`\dim_to_decimal_in_unit:nn` 4977
`\dim_use:N` 132, 3938
`\dim_zero:N` 2211
`\c_max_dim` 2124, 2455
`\c_zero_dim`
... 2132, 2144, 2159, 2175, 2182, 2369, 2446, 2935, 3630
`\directlua` 708, 709
`\do` 430
`\document` 5150, 5153

E

`\edef` 4664, 4665
`\else` 420, 423, 424, 428, 1697,
1698, 1702, 1703, 2506, 2515, 2524, 2540, 2545, 2581,
2586, 2667, 2675, 2691, 2697, 2703, 2716, 2726, 2786,
2791, 2810, 2815, 2836, 2841, 2850, 2872, 2888, 2901,
3438, 3466, 4676, 4677, 4703, 4727, 4776, 4808, 4840, 4889
else commands:
`\else:` 506, 935, 1890, 3044, 3260
`\em` 1695
`\eminnershape` 1697, 1698, 1703, 1705
`\end` 3320, 3339, 3358, 5039
`\endcsname` 419, 422, 423, 2793,
2802, 2803, 2804, 2811, 2831, 2832, 2834, 2837, 2846,
2847, 2848, 2851, 4671, 4673, 4674, 4675, 4678, 4680, 4684
`\endgroup` 420, 437, 2566, 2607, 2623, 2644, 2749,
2768, 2833, 2871, 2896, 3322, 3341, 3360, 4664, 4672, 4687
`\endinput` 420, 4669
`\endlinechar` 4643, 4644
`\everypar` 2790, 2891, 2903
`\EverySelectfont` 1953, 1956
exp commands:
`\exp_after:wN`
131, 853, 869, 871, 935, 940, 3125, 3761, 3808, 3843, 5323
`\exp_args:cc` 3371
`\exp_args:Nc` 3097, 3102, 5277

`\exp_args:Nco` 3207
`\exp_args:Ne` 1611
`\exp_args:NNc` 819, 5187
`\exp_args:Nnc` 105
`\exp_args:NNNo` 3937
`\exp_args:NNo` 2431
`\exp_args:Nnx` 3383
`\exp_args:No` 366, 563, 948, 1397, 3121, 4954, 5339
`\exp_args:Nx` 274, 591, 663, 2164, 3757, 3936
`\exp_last_unbraced:Ne` 5288
`\exp_last_unbraced:NNNNo` 134
`\exp_last_unbraced:NNo` 1944
`\exp_not:N` 485,
593, 594, 595, 888, 920, 1023, 1105, 1106, 1446, 1770,
1771, 1772, 1786, 2330, 2331, 2332, 2337, 2339, 2340,
2341, 2342, 2345, 2346, 2349, 2361, 2362, 2363, 2364,
2365, 2374, 2375, 2376, 2378, 2379, 2393, 2395, 2397,
2983, 2985, 2986, 2987, 2988, 3109, 3110, 3119, 3120,
3157, 3159, 3160, 3383, 3777, 4008, 5268, 5269, 5270, 5358
`\exp_not:n` 82, 130, 161,
1102, 1103, 1104, 1235, 1236, 1447, 1448, 1449, 1474,
1530, 1531, 1559, 1765, 1787, 1909, 2380, 3128, 3372,
5129, 5153, 5243, 5244, 5286, 5358, 5359, 5360, 5415, 5416
`\expandafter` 419, 420, 422, 423, 431, 435,
4646, 4647, 4671, 4672, 4673, 4674, 4675, 4680, 4684, 4685
`\ExplSyntaxOff`
.... 79, 85, 494, 686, 1696, 1946, 3230, 3303, 4983, 5221
`\ExplSyntaxOn` 84, 719, 3364, 4923

F

`\familydefault` 1768
`fancyhdr` 32, 326
`\fangsong` 4613
`\fi` 420, 423,
424, 432, 433, 434, 436, 2497, 2502, 2508, 2517, 2526,
2529, 2544, 2549, 2559, 2570, 2585, 2590, 2601, 2627,
2651, 2659, 2660, 2663, 2667, 2677, 2695, 2701, 2707,
2719, 2729, 2733, 2742, 2743, 2753, 2758, 2772, 2776,
2782, 2788, 2795, 2797, 2819, 2820, 2838, 2852, 2855,
2876, 2885, 2904, 2905, 3255, 3258, 3262, 3272, 3273,
3278, 3286, 3289, 3294, 3297, 3316, 3335, 3354, 3438,
3466, 4676, 4677, 4686, 4722, 4746, 4803, 4835, 4859, 4916
fi commands:
`\fi:` 541, 578, 597, 935, 1890, 1951, 3047, 3264, 3280
`\figurename` 2264
`figurename` 12
file commands:
`\g_file_curr_name_str` 678
`\file_if_exist:nTF` 48, 3987, 4002, 4390, 4537, 4540
`\file_if_exist_input:nTF` 100
`\file_input:n` 91, 101, 103, 526, 550
`fntef` 33, 34, 307
`\font` 4072, 4073, 4074, 4075, 4076, 4077
`fontset` 7, 239, 4021
`\fontsize` 3761

fontspec commands:

`\fontspec_gset_family:Nnn` 1103
`\fontspec_set_family:Nnn` 1483

fontspec internal commands:

`__fontspec_fontname_wrap:n` 60, 1045
`\footnotesep` 3937
`\footnotesize` 3874, 3913, 3937

fp commands:

`\fp_eval:n` 2167, 4973
`\fp_if_nan:nTF` 3721, 3748, 3934
`\fp_set:Nn` 3722, 3749
`\fp_use:N` 2104, 3936
`\c_nan_fp` 190
`\c_zero_fp` 2104

G

`GBK` 6, 34, 227
`\gdef` . 427, 433, 3009, 3010, 3011, 3012, 3015, 3016, 3017, 3018
`\global` 2671, 2890, 2893
`\globaljfont` 871
`\glueexpr` 3310, 3321, 3329, 3340, 3348, 3359

group commands:

`\group_begin:` 489, 726, 734, 819, 829,
855, 881, 919, 941, 987, 1050, 1101, 1478, 1556, 1574,
1591, 1764, 1913, 1993, 2336, 2344, 2409, 2933, 3033,
3166, 3225, 3937, 4949, 5147, 5187, 5245, 5253, 5262, 5317
`\group_end:` 493, 502, 731, 816, 819, 834, 859,
894, 923, 945, 991, 1065, 1108, 1484, 1560, 1587, 1598,
1776, 1944, 2013, 2347, 2351, 2431, 2937, 3048, 3189,
3229, 3937, 4960, 5147, 5187, 5265, 5312, 5340, 5374, 5398
`\group_insert_after:N` 891, 892
`\gtfamily` 1697, 1702

H

hbox commands:

`\hbox_set:Nn` 1861, 2926, 3218
`heading` 9, 280
`\heiti` 4613
`\hskip` 2828, 2844, 2866, 2873, 2899, 3286, 3289
`\hspace` 2539, 2580, 2958, 2966, 3149
`\Huge` 2604, 2641, 2746, 2765, 3442, 3473, 3891, 3930
`\huge` 2562,
2600, 2620, 2741, 3424, 3441, 3445, 3471, 3479, 3890, 3929
`hyperref` 33, 331
`\hypersetup` 336, 337, 1795
`\hyphenchar` 4072, 4073, 4074, 4075, 4076, 4077

I

`\if` 426
if commands:
`\if_bool:N` 503, 585
`\if_cs_exist:N` 558, 1912
`\if_false:` 123
`\if_meaning:w` 1889
`\if_true:` 122
`\ifctexpdf` 121
`\ifdim` 2780, 2823, 2861, 2880

`\ifnum` 424, 2535,
 2557, 2576, 2599, 2680, 2739, 2806, 2836, 2850, 3255,
 3258, 3262, 3269, 3270, 3278, 3286, 3289, 3294, 3297, 4676
`\ifodd` 421, 433, 2502, 2504, 2529, 2536, 2570,
 2577, 2627, 2651, 2673, 2684, 2733, 2753, 2758, 2772,
 2784, 2795, 2811, 2824, 2862, 2881, 2885, 3314, 3333, 3352
`\ifx` 419, 422, 423, 4674, 4675, 4680
`\ifzhmappdf` 4672, 4701, 4725, 4749, 4806, 4838, 4862
`\ignorespaces` 2906
`\immediate` 4683
`indent` 32, 209
`\indexname` 2267
`indexname` 12
`\input` 4922, 4986
`\insertcontinuationtext` 2276, 2285, 2292
`\insertpart` 3318, 3319
`\insertromanpartnumber` 3308, 3411
`\insertsection` 3337, 3338
`\insertsectionnumber` 3327, 3509
`\insertsubsection` 3356, 3357
`\insertsubsectionnumber` 3346, 3538
int commands:
`\int_case:nn` 391, 3848
`\int_compare:nNnTF`
 564, 2140, 2184, 3597, 3613, 3629, 3719, 3746
`\int_const:Nn` 1037, 1642, 1643
`\int_div_truncate:nn` 82, 571
`\int_eval:n` 1634, 1635, 1638, 1639
`\int_from_hex:n` 563
`\int_gincr:N` 1193, 1497
`\int_gset:Nn`
 .. 110, 170, 179, 180, 181, 285, 287, 355, 359, 3720, 3747
`\int_incr:N` 2980
`\int_mod:nn` 573
`\int_new:N` 61, 109, 169, 1500
`\int_set:Nn` 548, 554, 562, 2137, 2179, 2436
`\int_sub:Nn` 567, 2185
`\int_to_Hex:n` 565, 570, 572
`\int_use:N` 486, 554, 1195, 1498, 2984, 4954
`\int_zero:N` 2977
`\c_zero_int` 2140
int internal commands:
`__ctexpar` 2867
`\interlinepenalty` . 2554, 2596, 2616, 2637, 2761, 2829, 2867
iow commands:
`\iow_close:N` 4943
`\iow_indent:n` 5134
`\iow_new:N` 4945
`\iow_now:Nn` 4955
`\iow_open:Nn` 4940
`\itemsep` 3871, 3882, 3910, 3921
`\itshape` 1697, 1698, 1702, 1703

J
`\jfam` 979, 1038, 1692

K

`\kaishu` 4613
`\kanjifamily` 1682, 1683, 1684, 1686
`\kanjifamilydefault` 1681
kernel internal commands:
`\l__kernel_expl_bool` 83
keys commands:
`\l_keys_choice_tl` 275
`\keys_define:nn` 66, 69, 1042, 1384, 1494, 1648
`\keys_if_exist:nnTF` 1646
`\l_keys_key_tl` 148, 150, 164
`\keys_set:nn` 71, 73
`\keys_set_known:nn` 3677
`\keys_set_known:nnN` 1054

L

`\labelformat` 3383, 3385
`\languagealias` 3706, 3711, 3732, 3737
`\LARGE` 3889, 3928
`\Large` 2558, 3422, 3428, 3496, 3500, 3888, 3927
`\large` 3521, 3887, 3926
`\lastbox` 2894
`\leavevmode` 2497, 2776
left commands:
`\c_left_brace_str` 5309
`\leftmargin` 3868, 3879, 3907, 3918
`\leftmargini` 3868, 3879, 3907, 3918
`\let` 429, 432, 2809, 2818, 3861, 3900, 4667, 4673
`\linespread` 3936
`linespread` 11, 189
`linestretch` 11, 2109
`\linewidth` 2138, 2146, 2178
`\lishu` 4613
`\listfigurename` 2262
`listfigurename` 12
`\listtablename` 2263
`listtablename` 12
`\LoadClass` 399, 404, 409, 414
`\long` 4681
`\ltjdefcharrange` . 687, 688, 689, 691, 696, 697, 701, 704, 705
`LTJFONTUID` 1494
`\ltjgetparameter` 1866, 1893, 1971, 1985
`\ltjsetkanjiskip` 1670
`\ltjsetmathletter` 989
`\ltjsetparameter` 707, 713, 1667
`\ltjsetxkanjiskip` 1672

lua commands:

`\lua_now:n` 736, 1995
`\LuaTeXjversion` 803

M

`\MAKESPA` 4946, 4987
`\markboth` 2470, 2564, 2593
`\mathgroup` 979
max commands:
`\c_max_char_int` 1643
`\mcfamily` 1702

- \mddefault 1024, 1028, 1034, 1372, 1377
- mode commands:
 - \mode_if_math:TF 973
- msg commands:
 - \msg_critical:nnn 53, 678, 3961
 - \msg_error:nn 4041
 - \msg_error:nnn 19, 22, 38, 41, 669, 682, 685, 2249, 3762, 3844
 - \msg_error:nnnn 4007, 4036
 - \msg_fatal:nn 4928
 - \msg_new:nnn . 147, 149, 163, 165, 381, 672, 1095, 1161, 1243, 2280, 3651, 3962, 4015, 4050, 4926, 5130, 5166, 5229
 - \msg_new:nnnn 12, 30, 45, 115, 2251, 3764, 4017, 4044
 - \msg_warning:nn 232, 386, 1238, 2034, 2045, 2302
 - \msg_warning:nnn 152, 1156, 3670, 4028, 5136, 5165, 5228
 - \msg_warning:nnnn 160, 1091, 3997
- N**
- \newCJKfontfamily 1262
- \NewDocumentCommand
 - 1122, 1257, 1262, 1268, 1278, 1284, 1290, 1297, 1303, 1309, 1315, 1522, 1622, 1685, 2052, 2055, 2065, 2092, 2163, 2200, 2207, 2228, 2230, 2399, 3756, 3941, 4614, 4615, 4616, 4617, 4619, 4620, 4622, 4632, 4633
- \newfontfeature 1039, 1040, 1041
- \newpage 2653, 2658
- \ngostype 44
- \nobreak 2558, 2567, 2624, 2741, 2746, 2750, 2765, 2769, 2882, 3423, 3472
- \nobreakspace 2558, 2600
- nocap 32, 303
- \noexpand 4665, 4667, 4668, 4669
- nofonts 32
- noindent 32
- nopunct 32
- \normalem 1835
- \normalfont 1294, 1301, 1307, 1355, 1694, 2434, 2554, 2596, 2616, 2637, 2736, 2761, 2988
- \normalsize 3555, 3572, 3586, 3855, 3894, 3932, 3940
- nospace 32
- nozhmap 32
- \null 2527, 2656
- \numberline 2690, 2837, 2851, 2966, 2974, 3160, 3225
- O**
- \onecolumn 2522
- P**
- \pagestyle 3697
- \par 2461, 2498, 2558, 2564, 2565, 2600, 2604, 2606, 2620, 2622, 2641, 2643, 2741, 2746, 2748, 2765, 2767, 2777, 2832, 2870, 2882, 3309, 3310, 3318, 3321, 3328, 3329, 3337, 3340, 3347, 3348, 3356, 3359, 3401, 3423, 3434, 3452, 3460, 3472, 3512, 3514, 3546, 3548
- paragraph/afterskip 34
- paragraph/beforeskip 34
- paragraph/numbering 16
- paragraph/runin 21
- \parindent 2159, 2160, 2182, 2203, 2205, 2210, 2211, 2554, 2616, 2736, 2761, 3631
- \parsep 3870, 3871, 3881, 3882, 3909, 3910, 3920, 3921
- \part 2495
- part/beforeskip 34
- part/fixbeforeskip 34
- part/numbering 16
- part/pagestyle 21, 2409
- \partmark 2469, 2552, 2594
- \partname 2558, 2600, 3308, 3406
- \PassOptionsToClass 398, 403, 408, 413
- \PassOptionsToPackage 504, 1799, 2213, 5138
- \pdfmapline 4750, 4751, 4752, 4753, 4754, 4755, 4756, 4757, 4758, 4759, 4760, 4761, 4762, 4763, 4764, 4765, 4766, 4767, 4768, 4769, 4770, 4771, 4772, 4773, 4774, 4775, 4863, 4864, 4865, 4866, 4867, 4868, 4869, 4870, 4871, 4872, 4873, 4874, 4875, 4876, 4877, 4878, 4879, 4880, 4881, 4882, 4883, 4884, 4885, 4886, 4887, 4888
- \pdfoutput 424, 4676
- \pdfstringdefDisableCommands 3944
- \pingfang 4613
- prg commands:
 - \prg_do_nothing: 450, 983, 1472, 1557, 1911, 2021, 2950, 5350
 - \prg_generate_conditional_variant:Nnn 1148
 - \prg_new_conditional:Npnn 897, 1886
 - \prg_new_protected_conditional:Npnn 1135, 5320
 - \prg_return_false: ... 900, 1145, 1890, 1894, 1898, 5330
 - \prg_return_true: 899, 1138, 1143, 1890, 1894, 1898, 5335
- \primitive 5039
- \ProcessKeysOptions 378
- \proofname 2254, 2269
- proofname 13
- prop commands:
 - \prop_clear:N 1052
 - \prop_get:NnN 1230
 - \prop_get:NnNTF 1137, 1189, 1227, 1465, 1511, 3760, 3815, 3842
 - \prop_gpop:NnNTF 1080, 1084, 1592
 - \prop_gput:Nnn . 1058, 1059, 1105, 1217, 1580, 1616, 3777
 - \prop_gremove:Nn 1089
 - \prop_if_empty:NTF 1151, 1336, 1416
 - \prop_map_break:n 1350
 - \prop_map_inline:Nn 1348, 1424
 - \prop_new:N 1075, 1076, 1077, 1222, 1406, 1600, 1621, 3771
 - \prop_put:Nnn 1403
 - \prop_remove:Nn 1402
- \protect 2483, 2489, 2690, 2711, 2712, 2837, 2851, 2966, 2974, 3159, 3171, 3173, 3177, 3179, 3183, 3185, 3193, 3198, 3210, 3211, 5281, 5282
- \ProvideDocumentCommand 2469
- \ProvidesFile 4679, 4688, 4690, 4692, 4694, 4696, 4698
- \providetranslation 5044, 5045, 5046, 5047, 5048, 5049, 5050, 5051, 5052, 5053, 5054, 5055, 5056, 5057, 5058, 5059, 5060, 5061, 5062, 5063, 5064, 5065, 5066, 5067, 5068,

5069, 5070, 5071, 5072, 5073, 5074, 5075, 5076, 5077,
5078, 5079, 5080, 5081, 5082, 5083, 5084, 5085, 5086,
5087, 5088, 5089, 5090, 5091, 5092, 5093, 5094, 5095,
5096, 5097, 5098, 5099, 5100, 5101, 5102, 5103, 5104,
5105, 5106, 5107, 5108, 5109, 5110, 5111, 5112, 5113,
5114, 5115, 5116, 5117, 5118, 5119, 5120, 5121, 5122, 5123
punct 10, 32, 266, 2037
\punctstyle 582, 2042

Q

\quad 2802,
3255, 3256, 3258, 3259, 3262, 3263, 3275, 3279, 3287,
3290, 3295, 3298, 3429, 3480, 3489, 3522, 3556, 3573, 3587
quark commands:
 \q_mark 1425, 1433, 5324, 5327
 \q_stop
 132, 135, 934, 937, 940, 943, 5289, 5294, 5303, 5324, 5327
\quotation 3954

R

\raggedright 2554, 2616, 2736, 2761, 3421, 3470
\refname 2270, 2275, 2282, 2291, 2299, 2300
refname 13
\refstepcounter ... 2538, 2579, 2686, 2813, 3366, 3388, 3389
\relax 419, 422, 423, 426, 2576, 2599, 2778, 2814, 2822, 2828,
2844, 2860, 2873, 2879, 2898, 3286, 3289, 3310, 3321,
3329, 3340, 3348, 3359, 4639, 4643, 4646, 4674, 4675, 4680
\renewcommand 2496, 2512, 2666
\RenewDocumentCommand 3389
\RequirePackage
 27, 56, 57, 59, 312, 318, 329, 339, 505, 508, 509,
 510, 662, 680, 683, 721, 733, 1757, 1808, 1817, 2214, 3751
\RequirePackageWithOptions 5139
resetalternatfont 30, 1563
reverse commands:
 \reverse_if:N 585
\rmdefault 1770

S

scan commands:
 \scan_stop: 69, 132, 871, 979, 1558, 2130, 2138, 4950, 5149
scheme 10, 291
\scriptsize 3885, 3924
\secdef 2509, 2531, 2678
section/afterskip 34
section/beforeskip 34
section/numbering 16
section/runin 21
\sectionname 3327, 3506
\selectfont 1129, 1583, 1596, 1686, 1943, 1950, 1964, 2094, 3761
seq commands:
 \seq_const_from_clist:Nn . 456, 2311, 2321, 3023, 4930
 \seq_gpop:NNTF 847
 \seq_gpush:Nn 842
 \seq_gput_left:Nn 2317, 2318
 \seq_gput_right:Nn 1155, 3783
 \seq_gset_eq:NN 2316

\seq_if_in:NnTF 462, 1153
\seq_map_function:NN 3092, 3214
\seq_map_inline:Nn
 ... 2426, 2978, 3026, 3036, 3045, 3068, 3151, 3377, 4953
\seq_new:N 850, 1160, 2315, 3772
\seq_use:Nnnn 3769
\setbox 2894
\setCJKfamilyfont 1174, 1262, 4201,
 4203, 4205, 4207, 4243, 4244, 4245, 4246, 4265, 4267,
 4269, 4271, 4314, 4316, 4318, 4320, 4333, 4334, 4335,
 4336, 4337, 4338, 4379, 4381, 4382, 4383, 4384, 4385,
 4413, 4414, 4415, 4416, 4417, 4418, 4458, 4459, 4460,
 4461, 4472, 4473, 4474, 4475, 4476, 4477, 4478, 4494,
 4496, 4498, 4531, 4532, 4533, 4554, 4555, 4556, 4557,
 4558, 4560, 4561, 4604, 4605, 4606, 4607, 4608, 4609, 4610
\setCJKmainfont 1170, 1290, 4191, 4239, 4255,
 4304, 4329, 4375, 4405, 4454, 4464, 4488, 4528, 4549, 4601
\setCJKmathfont 1290
\setCJKmonofont 1172, 1290, 4199, 4242, 4263,
 4312, 4332, 4378, 4412, 4457, 4471, 4492, 4530, 4553, 4603
\setCJKromanfont 1296, 1322
\setCJKsansfont 1171, 1290, 4197, 4241, 4261,
 4310, 4331, 4377, 4411, 4456, 4470, 4490, 4529, 4551, 4602
\SetSymbolFont 1035, 1376, 1380, 1691
\sfdefault 1771
\shapedefault
 ... 1024, 1028, 1030, 1034, 1036, 1372, 1374, 1377, 1381
skip commands:
 \skip_horizontal:N 1876, 2936
 \skip_if_eq:nnTF 1893, 1897, 1970, 2480, 2486
 \skip_new:N 1885, 1983, 2465
 \skip_set:Nn 1866, 1869,
 1974, 1984, 2131, 2142, 2187, 2195, 2467, 2479, 2485, 2934
 \skip_set_eq:NN 1883, 1976
 \skip_sub:Nn 2456, 2463
 \skip_use:N 2483, 2489
 \skip_zero:N 2151
 \c_zero_skip 2367, 2368, 2417, 2418, 2480, 2486
\small 3863, 3902
\songti 4613
\space 424, 2687, 2741, 3406, 3468, 3506, 3537, 3638
space 10, 32, 271, 2016
\special 1724, 1725, 1728, 1729, 1737, 1738,
 1741, 1742, 1815, 1829, 4704, 4705, 4706, 4707, 4708,
 4709, 4710, 4711, 4712, 4713, 4714, 4715, 4716, 4717,
 4718, 4719, 4720, 4721, 4728, 4729, 4730, 4731, 4732,
 4733, 4734, 4735, 4736, 4737, 4738, 4739, 4740, 4741,
 4742, 4743, 4744, 4745, 4777, 4778, 4779, 4780, 4781,
 4782, 4783, 4784, 4785, 4786, 4787, 4788, 4789, 4790,
 4791, 4792, 4793, 4794, 4795, 4796, 4797, 4798, 4799,
 4800, 4801, 4802, 4809, 4810, 4811, 4812, 4813, 4814,
 4815, 4816, 4817, 4818, 4819, 4820, 4821, 4822, 4823,
 4824, 4825, 4826, 4827, 4828, 4829, 4830, 4831, 4832,
 4833, 4834, 4841, 4842, 4843, 4844, 4845, 4846, 4847,
 4848, 4849, 4850, 4851, 4852, 4853, 4854, 4855, 4856,
 4857, 4858, 4890, 4891, 4892, 4893, 4894, 4895, 4896,

4897, 4898, 4899, 4900, 4901, 4902, 4903, 4904, 4905,
4906, 4907, 4908, 4909, 4910, 4911, 4912, 4913, 4914, 4915
\SplitArgument 1523, 1623, 2400

str commands:

\c_backslash_str 5308
\c_colon_str 966, 967, 968
\c_dollar_str 4537
\str_case:nnTF 1768, 5279, 5294
\str_case_e:nnTF 1168, 1179
\str_const:Nn 43, 50
\str_if_empty:NTF 5365, 5401, 5410
\str_if_eq:nnTF 507, 581, 586, 1012, 1340,
1804, 1812, 3702, 3729, 3743, 4027, 4030, 4047, 4052, 5285
\str_if_eq_p:nn 3994, 3995
\str_lowercase:n 481
\str_new:N 5314, 5315, 5316
\str_set:Nn 5332, 5333, 5334
\strutbox 3938
sub3section 9, 283
sub4section 9, 283
subparagraph/afterskip 34
subparagraph/beforeskip 34
subparagraph/numbering 16
subparagraph/runin 21
subsection/afterskip 34
subsection/beforeskip 34
subsection/numbering 16
subsection/runin 21
\subsectionname 3346, 3537
subsubsection/afterskip 34
subsubsection/beforeskip 34
subsubsection/numbering 16
subsubsection/runin 21
\symmincho 1692

sys commands:

\c_sys_engine_str 44
\sys_if_engine_pdftex:TF ... 229, 315, 385, 3966, 3972
\sys_if_engine_pdftex_p: 4625
\sys_if_engine_uptex:TF 3968, 3974
\sys_if_engine_uptex_p: 4626
\sys_if_engine_xetex:TF 309, 4924
\sys_if_output_pdf:TF 121, 491, 1807, 3973
\sys_if_platform_windows:TF 3978

T

\tablename 2265
tablename 12
\TeX 3942

T_EX and L^AT_EX 2_ε commands:

\@italiccorr 1668
\@par 2829, 3490, 3523, 3557, 3601, 3617
\@afterheading 2572, 2629, 2718, 2728, 2887
\@afterindentfalse 2507, 2672, 2676, 2781, 2787
\@afterindenttrue 2505, 2674, 2779, 2785
\@Alph 3634, 3639
\@break@tfor 432

\@chapapp 2687, 2741, 3271, 3294
\@chapter 2678, 2679
\@chinese 2216, 2225
\@classoptionslist 45, 427, 433, 435
\@clubpenalty 2902
\@currentHref 3057
\@currentlabel 3368, 3372
\@currentx 4, 138, 5129, 5195, 5201
\@currname 4, 138, 5129, 5196, 5202
\@dblarg 2800
\@defaultfamilyhook 540, 1332
\@empty 105, 2809, 2818
\@endpart 2608, 2645, 2647
\@EverySelectfont@Init 1912, 1926, 1938
\@firstofone 54
\@firstoftwo 3210
\@gobble 54
\@hangfrom 2828, 2866
\@ifclassloaded 3647
\@ifpackagelater 37, 40, 668, 681, 684, 1842
\@ifpackageloaded 18, 21, 677, 1792, 3131, 3236, 5160, 5173
\@ifpackagewith 3076, 3085, 3136
\@ifstar 2798
\@listI 3861, 3900
\@listi 3861, 3868, 3879, 3900, 3907, 3918
\@M 2554, 2596, 2616, 2637, 2761, 2829, 2867, 2895
\@makechapterhead 2715, 2717, 2730
\@makeschapterhead 2725, 2727, 2755
\@minus 3491, 3524, 3558, 3574, 3588, 3857,
3859, 3865, 3867, 3869, 3870, 3876, 3878, 3880, 3881,
3896, 3898, 3904, 3906, 3908, 3909, 3915, 3917, 3919, 3920
\@namedef . 3171, 3173, 3177, 3179, 3183, 3185, 3193, 3198
\@nameuse 3171, 3173, 3183, 3185, 3193
\@ne 429, 3129, 3258, 3289
\@nil 920
\@nobreakfalse 2889
\@noskipsecfalse 2893
\@noskipsectrue 2890
\@onlypreamble 39, 453, 518, 1318,
1319, 1320, 1321, 1322, 1323, 1753, 1754, 1755, 1756, 4020
\@part 2509, 2531, 2533
\@pkgextension 6, 7, 8, 9, 20, 23, 679, 5195, 5201
\@plus 3435,
3436, 3491, 3492, 3524, 3525, 3558, 3559, 3574, 3588,
3602, 3618, 3857, 3858, 3859, 3865, 3866, 3867, 3869,
3870, 3876, 3877, 3878, 3880, 3881, 3896, 3897, 3898,
3904, 3905, 3906, 3908, 3909, 3915, 3916, 3917, 3919, 3920
\@popfilename 138, 5198
\@pushfilename 138
\@reset@ptions 138, 5192
\@rmfamilyhook 537, 1329, 1682
\@schapter 2678, 2720
\@seccntformat 2801, 2814
\@secondoftwo 3211
\@secpenalty 2792, 3494, 3527, 3561, 3575, 3589
\@sect 2800, 2805

\@setfontsize	3847	\CJK@input	49, 542
\@sffamilyhook	538, 1330, 1683	\CJK@loadBinding	48, 49, 589
\@spart	2509, 2531, 2610	\CJK@loadEncoding	48
\@ssect	2799, 2857	\CJK@makeActive	49, 588
\@startsection	97, 99, 106, 2775, 2983	\CJK@plane	48, 79, 470, 471, 481, 1953
\@svsec	2809, 2814, 2818, 2828, 2830, 2844, 2845	\CJK@surr	48, 558
\@svsechd	2842, 2873, 2874, 2896	\CJK@upperReset	49
\@tempdima	3219, 3221	\CJKfamily	61
\@tempskipa	2778, 2780, 2781, 2792, 2822, 2823, 2860, 2861, 2879, 2880, 2883, 2898, 2899	\CJKfamilydefault	49, 49, 65, 65, 72, 74, 74, 74, 74
\@tempswafalse	2525	\CJKglue	82, 116
\@tempswatrue	2523	\CJKhook	50, 50
\@tfor	430	\CJKrmdefault	47
\@topnewpage	2715, 2725	\CJKsymbol	79
\@topnum	2671	\CJKunderdotbasesep	33
\@ttfamilyhook	539, 1331, 1684	\CJKunderline	33
\@xsect	2856, 2877, 2878	\clearpage	49
\@zhdig	2224	\contentsname	12, 12
\@zhnum	2216, 2223	\CS	56
\abstractname	12	\CTEX@add	429, 432, 433
\addCJKfontfeature	69	\CTEX@addloflotskip	2476, 2713
\AfterEndPreamble	137	\CTEX@addtocline	2493, 2551, 2592, 2709, 2840, 2854
\algorithmname	13	\CTEX@afterindent	2784, 2944, 2951
\appendixname	12, 25	\CTEX@aftertitle	2870, 2875, 2943, 2950
\AtBeginDocument	39, 49	\CTEX@appendix@number	2998, 3010, 3016
\AtBeginDvi	46	\CTEX@appendix@numbering	2999, 3012, 3018
\AtBeginShipoutFirst	46	\CTEX@beginallalchar	722, 724
\AtEndOfClass	40	\CTEX@chapter@afterindent	2673
\AtEndOfPackage	40	\CTEX@chapter@aftername	2745
\AtEndOfPackageFile*	137	\CTEX@chapter@afterskip	2752, 2771
\AtEndPreamble	137	\CTEX@chapter@aftertitle	2748, 2767
\baselineskip	11, 11, 79, 82	\CTEX@chapter@beforeskip	2732, 2757
\bfseries	50	\CTEX@chapter@break	2669
\bibname	12	\CTEX@chapter@fixskip	2733, 2753, 2758, 2772
\c@chapter	3639	\CTEX@chapter@format	2738, 2763
\c@secnumdepth	2535, 2557, 2576, 2599, 2680, 2739, 2806, 2836, 2850, 3255, 3258, 3262, 3269, 3270, 3278, 3286, 3289, 3294, 3297	\CTEX@chapter@lofskip	2415
\c@section	3634	\CTEX@chapter@lotskip	2416
\caption	33	\CTEX@chapter@numbering	2684, 3018
\catcode	38, 47, 48, 128, 138	\CTEX@chapter@pagestyle	2414, 2670
\ccwd	11, 11, 27, 27, 33, 33, 39, 77, 78, 78, 82, 82, 82, 83	\CTEX@chapter@titleformat	2747, 2766
\chapter	9, 16, 21, 25, 25, 26, 89, 96, 110	\CTEX@chapter@tocline	2955
\chaptermark	13	\CTEX@chaptername	2745
\chaptername	17	\CTEX@document@left@hook	5146, 5152
\char	52	\CTEX@document@right@hook	5148, 5154
\CheckCommand	78	\CTEX@endallalchar	723, 730
\Chinese	33	\CTEX@Family@CMap	467, 495, 498
\chinese	28, 33, 84	\CTEX@fixheadingskip	2454, 2459, 2502, 2570, 2627, 2651, 2753, 2772, 2795, 2885
\CJK@ignorespaces	50, 598	\CTEX@fixskip	2795, 2885, 2945, 2952
\CJK@encoding	60, 61, 68, 826, 830, 914, 1019, 1073, 1371, 1374, 1376, 1380, 1520, 1530, 1531, 1549, 1550	\CTEX@fixtopskip	2451, 2529, 2733, 2758
\CJK@envStart	49, 49, 593	\CTEX@getttitle	2550, 2591, 2614, 2635, 2708, 2723, 2821, 2859, 3063
\CJK@family	55, 817, 831, 882, 914, 1126, 1582, 1592, 1594, 1964, 1965	\CTEX@hang	2914, 2946, 2953
\CJK@ignorespaces	50, 50, 602, 606	\CTEX@heading@format@initial	2432, 2555, 2597, 2617, 2638, 2737, 2762, 2826, 2864, 3311, 3330, 3349
		\CTEX@heading@glue	2900, 2931
		\CTEX@headinghang	2560, 2602, 2619, 2640, 2744, 2764, 2907

\CTEX@headingskip	2468, 2503, 2530, 2571, 2628, 2652, 2734, 2754, 2759, 2773, 2796, 2886
\CTEX@hyperheadinghook	3031, 3061
\CTEX@ifnamefalse	2471, 2541, 2546, 2582, 2587, 2612, 2633, 2692, 2698, 2704, 2721, 2807, 2816, 2858
\CTEX@ifnametrue	2471, 2537, 2578, 2685, 2812
\CTEX@indentbox	2449, 2845, 2874, 3313, 3332, 3351
\CTEX@makeanchor	2542, 2547, 2583, 2588, 2693, 2817, 3021, 3029, 3041, 3046, 3052
\CTEX@makeanchor@chapter	2699, 2705
\CTEX@makeanchor@schapter	2722
\CTEX@makeanchor@sect	2808
\CTEX@makeanchor@spart	2613, 2634
\CTEX@makeanchor@ssect	2799
\CTEX@part@afterindent	2504
\CTEX@part@aftername	2561, 2603, 3315
\CTEX@part@afterskip	2569, 2626, 2650, 3321
\CTEX@part@aftertitle ...	2565, 2606, 2622, 2643, 3319
\CTEX@part@beforeskip	2501, 2528, 3310
\CTEX@part@break	2499, 2518
\CTEX@part@fixskip	2502, 2529, 2570, 2627, 2651
\CTEX@part@format	2556, 2598, 2618, 2639, 3312
\CTEX@part@indent	3313
\CTEX@part@numbering	2536, 2577, 3314
\CTEX@part@pagestyle	2413, 2520
\CTEX@part@titleformat ..	2563, 2605, 2621, 2642, 3319
\CTEX@part@tocline	2955
\CTEX@partname	2561, 2603, 3315
\CTEX@popfilename@hook	5198
\CTEX@postappendix	3003, 3011, 3017
\CTEX@postchapter	3017
\CTEX@postsection	3011
\CTEX@preappendix	3002, 3009, 3015
\CTEX@prechapter	3015
\CTEX@presection	3009
\CTEX@reset@ptions@hook	5192
\CTEX@runin	2824, 2862, 2881, 2947, 2954
\CTEX@save@appendix	3004, 3007
\CTEX@save@refstepcounter	3388, 3391
\CTEX@section@aftername	3334
\CTEX@section@afterskip	3340
\CTEX@section@aftertitle	3338
\CTEX@section@beforeskip	3329
\CTEX@section@format	3331
\CTEX@section@indent	3332
\CTEX@section@numbering	3012, 3333
\CTEX@section@titleformat	3338
\CTEX@sectionhang	2830, 2868, 2907
\CTEX@sectionname	3334
\CTEX@selectfont@hook	1906, 1953, 1956
\CTEX@selectfont@save	1917, 1932, 1936, 1939
\CTEX@setcurrentlabel@n	3366, 3392
\CTEX@setheadingskip	2466, 2501, 2528, 2569, 2626, 2650, 2732, 2752, 2757, 2771, 2794, 2884
\CTEX@subsection@aftername	3353
\CTEX@subsection@afterskip	3359
\CTEX@subsection@aftertitle	3357
\CTEX@subsection@beforeskip	3348
\CTEX@subsection@format	3350
\CTEX@subsection@indent	3351
\CTEX@subsection@numbering	3352
\CTEX@subsection@titleformat	3357
\CTEX@subsectionname	3353
\CTEX@thechapter	3016
\CTEX@thesection	3010
\CTEX@titleformat@n	97, 2869, 2875, 2942, 2949
\CTEX@titleslabel@clear	3180, 3190
\CTEX@titleslabel@set	3174, 3186, 3190
\CTEX@toc@width@n	3216, 3231, 3243
\CTEX@todayold	2232, 2237
\CTEX@update@sectionformat@n	2783, 2940
\CTEX@verbatim@font@hook	1662, 1666
\ctex@zhmap@endinput	4918
\CTEXdigits	28
\CTEXifname	24, 26
\CTEXnumber	28, 28
\CTEXnumberline	24
\CTEXoptions	33, 33
\ctexset	5, 5, 6, 8, 12, 15, 29, 30, 33, 33, 117
\CTEXsetup	33
\CTEXthechapter	13, 14, 25, 101
\CTEXthesection	13, 17, 104
\CTEXunderdotbasesep	33
\CTEXunderline	33
\curr@fontshape	832, 866, 875, 879, 921
\DeclareAlternateKanjiFont	68
\DeclareFontEncoding	58
\DeclareFontFamily	46, 56
\DeclareFontShape	56
\DeclareFontShape@	68, 1481
\DeclareRobustCommand	139
\define@newfont	56, 858
\do@subst@correction	56, 57, 857
\em	72
\escapechar	58
\EverySelectfont	78
\external@font	868, 869, 871
\extract@font	56, 856
\f@baselineskip	2130
\f@encoding	830, 874, 976
\f@family	55, 831, 874, 882
\f@series	826, 914
\f@shape	826, 914
\f@size	826, 832, 909, 917, 920, 921, 1862, 1903, 3222
\fangsong	8
\fbox	18
\figurename	12
\font	54, 56
\font@name	55, 56, 835, 842, 843, 848, 853, 871, 872, 889, 957, 958, 959, 960, 961, 968
\fontcharwd	78
\footnotesep	115, 115

\footnotesize	115	\ltj@tempcntc	58, 930
\g@addto@macro	537,	\ltjalchar	52
538, 539, 540, 1329, 1330, 1331, 1332, 1682, 1683, 1684		\ltjsetkanjiskip	72
\get@external@font	865	\ltjsetparameter	72
\getanddefine@fonts	946	\ltjsetxkanjiskip	72
\globaldefs	55	\m@ne	2535, 2557, 2680, 2739,
\globaljfont	56	3262, 3269, 3270, 3294, 4072, 4073, 4074, 4075, 4076, 4077	
\H@old@chapter	3035	\markboth	13
\hbox	22, 89	\markright	13
\heiti	8, 127	\math@bgroup	975
\hrule	89	\math@egroup	980
\Hy@chapapp	2693, 2699, 2705, 2722	\math@fonts	963
\Hy@driver	1812	\mathbb	116
\Hy@MakeCurrentHrefAuto	3054	\maxdimen	11, 82
\Hy@org@chapter	3035	\meaning	139, 140, 140, 141
\Hy@raisedlink	3055	\newcommand	139
\Hy@unicodetrue	75	\newrobustcmd	139
\hyper@anchorend	3058	\newtitlemark	102
\hyper@anchorstart	3057	\nfss@catcodes	60
\HyPsd@ConvertToUnicode	75	\noindent	88
\HyPsd@LoadUnicode	76	\normalsize	8, 112
\HyPsd@pdfencoding	76	\nouppercase	48
\HyPsd@ToBigChars	76	\NR@getttitle	3067
\if@mainmatter	2682, 2740, 3269, 3665, 3666	\numberline	102
\if@nobreak	2789	\p@	1862, 1903, 2600, 2711, 2712, 2731,
\if@noskipsec	2497, 2776, 2892	2741, 2751, 2756, 2770, 3222, 3434, 3462, 3463, 3464,	
\if@openright	2513, 2655, 2667, 3438, 3466	3465, 3472, 3857, 3858, 3859, 3865, 3866, 3867, 3869,	
\if@tempswa	2661	3870, 3876, 3877, 3878, 3880, 3881, 3896, 3897, 3898,	
\if@twocolumn	2521, 2714, 2724	3904, 3905, 3906, 3908, 3909, 3915, 3916, 3917, 3919, 3920	
\if@twoside	2654, 3253, 3276	\pagenumbering	28
\iffalse	58	\pagestyle	10
\ifHy@implicit	3034	\par@update	78, 1923, 1927, 1931
\ifin@	58, 432, 935	\paragraph	9, 21, 39
\iftrue	58	\parindent	11, 33, 82, 83, 88, 96, 116
\in@	431	\parskip	89
\indexname	12	\part	15, 21, 22, 23, 89, 96
\insertcontinuationtext	13	\partmark	89
\itdefault	57	\partname	17
\itshape	51	\partpage	15, 104
\jfam	72	\PassOptionsToClass	45
\kaishu	8	\PassOptionsToPackage	75
\kanjiskip	79	\pdfmapline	120
\labelformat	105, 105, 106	\pdfstringdef	75
\leftmark	13	\pickup@font	55
\linewidth	82	\pickup@jfont	862
\lishu	8	\pingfang	8, 127
\listfigurename	12	\ProcessKeysOptions	31, 39
\listtablename	12	\proofname	13, 85
\ltj@@does@alt@set	57	\protected	102
\ltj@@getjfontnumber	58, 929	\protected@edef	2229, 2231, 2814, 3368
\ltj@@set@stackfont	966, 967, 968	\ProvidesFile	129
\ltj@allalchar	728	\ps@fancy	3282
\ltj@curjfnt	56, 79	\ps@headings	3252
\ltj@pickup@altfont@auxy	54, 915	\refname	12, 13
\ltj@pickup@altfont@copy	54, 54, 927	\refstepcounter	105
\ltj@setpar@global	72, 965, 1670, 1672	\relax	48, 82

\rightmark	13	\x@protect	5282
\rmfamily	51	\xdef	102
\roman	28	\XeTeXglyphbounds	133
\scantokens	139	\xkanjiskip	79
\scr@new@selectfont	1920, 1935	\yahei	8, 127
\scr@selectfont	1932, 1936	\youyuan	8
\scriptfont@name	960, 967	\z@	713, 716, 2554, 2616, 2671, 2736, 2761, 2780, 2823, 2861, 2880, 2894, 3129, 3255, 3278, 3286, 3297, 3858, 3866, 3877, 3897, 3905, 3916
\section	9, 15, 16, 16, 17, 21, 22, 22, 23, 23, 23, 25, 25, 34, 86, 96, 103	\zhdig	84
\section*	16	\zhdigits	28
\sectionmark	13	\zhnum	84
\sectionname	17	\zhnumber	28, 28
\sectionpage	15, 104	\zihao	27
\selectfont	57, 74, 78, 78, 79, 82	\ziju	27
\setCJKfamilyfont	29, 29	\zw	78
\setCJKmainfont	29	tex commands:	
\sf@size	959	\tex_def:D	5358
\sffamily	51	\tex_dimexpr:D	132, 2138
\size@update	82, 82, 1947, 1948, 2094, 2108	\tex_divide:D	82, 2139, 2180
\sldefault	57	\tex_edef:D	5412
\songti	8	\tex_endlinechar:D	548, 554
\special	6, 46, 75	\tex_font:D	485, 4950
\split@name	920	\tex_glueexpr:D	2130
\SplitArgument	87	\tex_hangindent:D	2927
\ssf@size	961	\tex_iftrue:D	3666
\strutbox	115	\tex_ignorespaces:D	
\subparagraph	9, 21, 39	1123, 1282, 1287, 2027, 2164, 3757, 5149
\subsectionname	17	\tex_immediate:D	482
\subsectionpage	15, 105	\tex_indent:D	2447
\subst@correction	885, 892	\tex_interlinepenalty:D	2436
\tablename	12	\tex_kanjiskip:D	1869, 1883, 1897
\textfont@name	958, 966	\tex_noindent:D	2437, 2925
\tf@size	957	\tex_pagegoal:D	2455
\thechapter	13, 25, 25	\tex_parindent:D	2441, 2446, 2917
\thesection	13, 17, 25, 104	\tex_parskip:D	2463
\titleformat	99	\tex_pdffontattr:D	485
\titlespacing	99	\tex_pdflastobj:D	486
\today	11, 11	\tex_pdfobj:D	482
\trans@languagepath	3714, 3740	\tex_prevdepth:D	2462
\ttfamily	51	\tex_topskip:D	2456
\ttl@a	3149, 3157	\tex_XeTeXcharglyph:D	4954
\ttl@chapterout	3144	\tex_XeTeXglyphbounds:D	4978
\ttl@extract	99	\tex_xkanjiskip:D	1972, 1976, 1986
\ttl@setifthe	3203	\the	4643, 4646, 4647, 4666, 4670
\ttl@setsubmark	101, 3176, 3182	\thechapter	2687, 2690, 2741, 3271, 3294, 3469
\ttl@settopmark	101, 3170, 3201	\theparagraph	3571
\ttl@tocpart	3146, 3148	\thepart	2539, 2558, 2580, 2600, 3408
\ttlh@hang	3110, 3120	\thesection	3255, 3262, 3278, 3286, 3297, 3488
\ttlh@runin	3109, 3119	\thesubparagraph	3585
\tw@	432	\thesubsection	3258, 3289, 3520
\uppercase	48	\thesubsubsection	3554
\use@mathgroup	971, 1002	\thispagestyle	2519, 2520, 2657, 2668, 2670
\usepackage	32, 33, 34	\tiny	3886, 3925
\verbatim@font	1660, 1664	tl commands:	
\vskip	22	\c_space_tl	486
\wrong@fontshape	55		

- \tl_clear:N 201, 367, 1614,
1618, 1837, 1849, 1965, 2085, 2406, 2424, 2435, 3812, 5366
 - \tl_clear_new:N 60, 344
 - \tl_const:Nn 3, 397, 402, 407, 412, 533, 657, 679, 1019,
1032, 1325, 1326, 1327, 1328, 1370, 1677, 1678, 1679,
1680, 2104, 2328, 2334, 3648, 3667, 3669, 4538, 4541, 4542
 - \tl_gput_right:Nn 441, 463, 5143, 5145, 5182
 - \tl_gset:Nn 230, 233, 236, 387, 565,
568, 957, 959, 961, 1344, 1766, 3979, 3982, 3983, 3999, 4032
 - \tl_gset_eq:NN 1789
 - \tl_gset_rescan:Nnn 1351
 - \tl_head:N 3816
 - \tl_head:n 1395
 - \tl_if_blank:nTF 1429, 1438, 1638, 1639, 1654, 2057, 2067
 - \tl_if_empty:NTF 383, 821, 2157, 2340, 3382, 3990
 - \tl_if_empty:nTF 136, 1712, 1726, 1739, 5329
 - \tl_if_eq:NNTF 1762, 2097, 3727
 - \tl_if_exist:NTF
. 98, 532, 534, 535, 536, 1324, 1326, 1327, 1328, 1676,
1678, 1679, 1680, 2254, 2285, 2296, 2299, 3660, 3750, 5181
 - \tl_if_in:NnTF 5393
 - \tl_if_novalue:nTF 1250, 1271, 1632, 2405
 - \tl_map_inline:Nn 1544
 - \tl_map_inline:nn 3813
 - \tl_new:N 137, 139, 452,
534, 535, 536, 823, 824, 912, 1016, 1067, 1094, 1111,
1112, 1133, 1486, 1620, 1782, 1979, 2105, 2118, 2256,
2326, 2327, 3002, 3003, 5156, 5157, 5181, 5369, 5370, 5371
 - \tl_put_left:Nn 1938, 5198
 - \tl_put_right:Nn 351, 963,
1627, 1926, 1927, 1931, 2356, 2395, 2397, 3816, 3817, 5192
 - \tl_remove_all:Nn 1396
 - \tl_replace_all:Nnn 1442
 - \tl_rescan:nn 141, 5411, 5420
 - \tl_set:Nn 127, 196, 481, 513,
514, 515, 561, 825, 868, 913, 1013, 1014, 1053, 1116,
1128, 1270, 1439, 1441, 1482, 1681, 1784, 1916, 1964,
1980, 2041, 2081, 2119, 2167, 2257, 2404, 2407, 2411,
3106, 3125, 3144, 3149, 3157, 3726, 3944, 4577, 4578, 4579
 - \tl_set_eq:NN
... 830, 831, 882, 958, 960, 1117, 1142, 1582, 1594, 2106
 - \tl_set_rescan:Nnn 5367
 - \tl_tail:N 3111
 - \tl_tail:n 3121
 - \tl_to_str:n 135, 5310, 5339, 5380, 5381
 - \tl_trim_spaces:n 1441, 1611
 - \tl_use:N 443
 - \today 2232, 2237, 2240, 2245
 - today 11, 2232
 - token commands:
 - \c_parameter_token 102, 3234, 3236, 3239
 - \token_to_meaning:N 5324
 - \token_to_str:N 58, 161,
940, 948, 957, 959, 961, 1170, 1171, 1172, 1174, 1181,
1182, 1183, 1245, 2282, 2283, 3765, 4957, 5134, 5169, 5228
 - \toks 4643, 4644, 4646, 4647, 4666, 4670
 - \topsep 3869, 3880, 3908, 3919
 - \TrimSpaces 2055
 - \ttdefault 1772
 - \twocolumn 2662
 - \typeout 2687, 2688, 3144
- ## U
- ubuntu 34
 - um commands:
 - \um_input_math_symbol_table: 995, 999
 - \um_sym:nnn 998
 - um internal commands:
 - _um_input_math_symbol_table: 1007
 - _um_switchto_literal: 1004
 - _um_sym:nnn 1006
 - \undefined 4668
 - \unless 2824, 2862, 2881
 - \unskip 2897
 - \upshape 1702, 1705
 - use commands:
 - \use:N 832, 874, 875, 921, 957, 959, 961, 1037,
1086, 1595, 2479, 2485, 2494, 2911, 2974, 5394, 5414, 5428
 - \use:n 77, 140, 920,
1021, 1200, 1253, 1259, 1341, 1345, 1444, 1783, 2920,
2949, 3227, 3658, 3775, 4004, 5174, 5263, 5305, 5356, 5377
 - \use:nn 5422
 - \use_i:nn ... 124, 935, 2473, 5209, 5214, 5234, 5239, 5249
 - \use_i:nnn 251
 - \use_ii:nn 125, 935, 2471, 2475, 5209, 5214, 5234, 5239, 5250
 - \use_ii:nnn 141, 256
 - \use_iii:nnn 261
 - \use_none:n ... 197, 202, 207, 899, 1102, 1344, 1479, 2218
 - \use_none:nn 498
 - \usebeamercolor 3308, 3327, 3346, 3451, 3511, 3545
 - \usebeamerfont 3307, 3318,
3326, 3337, 3345, 3356, 3450, 3453, 3510, 3513, 3544, 3547
 - \uselanguage 3705, 3710, 3731, 3736
 - \usepackage 5134
 - UTF8 6, 34, 227
- ## V
- \verse 3954
 - \vfil 2527, 2649
 - \vskip 2568, 2571,
2600, 2625, 2628, 2652, 2741, 2751, 2754, 2770, 2773,
2883, 2886, 3309, 3328, 3347, 3434, 3452, 3472, 3512, 3546
 - \vspace 2530, 2731, 2734, 2756, 2759
- ## W
- windows 34
 - windowsnew 34
 - windowsold 34
 - winfonts 32
 - \write 4683

X

\x 425, 431, 435, 4645, 4648,
4649, 4650, 4651, 4652, 4653, 4654, 4655, 4656, 4657,
4658, 4659, 4660, 4661, 4662, 4663, 4664, 4671, 4681, 4685

x commands:

\x: 430
\xdef 4684
\xeCJKsetup 663, 1845, 2022, 2025, 2028, 2043

Y

\yahei 4613
\youyuan 4613

Z

\zhdig 2224
\zhdigits 28, 2231
zhmap 6, 32, 246
\znum 2223

znum commands:

\znum_counter:n 2215
\zhnumber 28, 2229
\znumsetup 2241, 2246
\zhtoday 2240, 2245
\zihao 27, 3756, 3765
zihao 8, 176
\ziju 27, 2163
\zw 714, 1866, 1904, 1981